Fundamental Tidy Data Science in R

Muhammad Aswan Syahputra
R Academy, Telkom University, Bandung
26 - 27 July 2019

- Sensory Scientist @ Sensolution.ID
- Using R for 4+ years, keen on Data Carpentry
- Initiator of Komunitas R Indonesia
- 📦: sensehubr, nusandata, bandungjuara, prakiraan, etc
- 🖥️: sensehub, thermostats, aquastats, bcrp, bandungjuara, etc

aswansyahputra

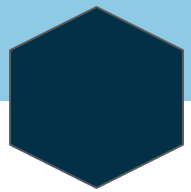
Know your neighbour!

- Who are you?
- What you do with data?
- How would you describe your experience with R?

Our goal

Artwork by @allison_horst

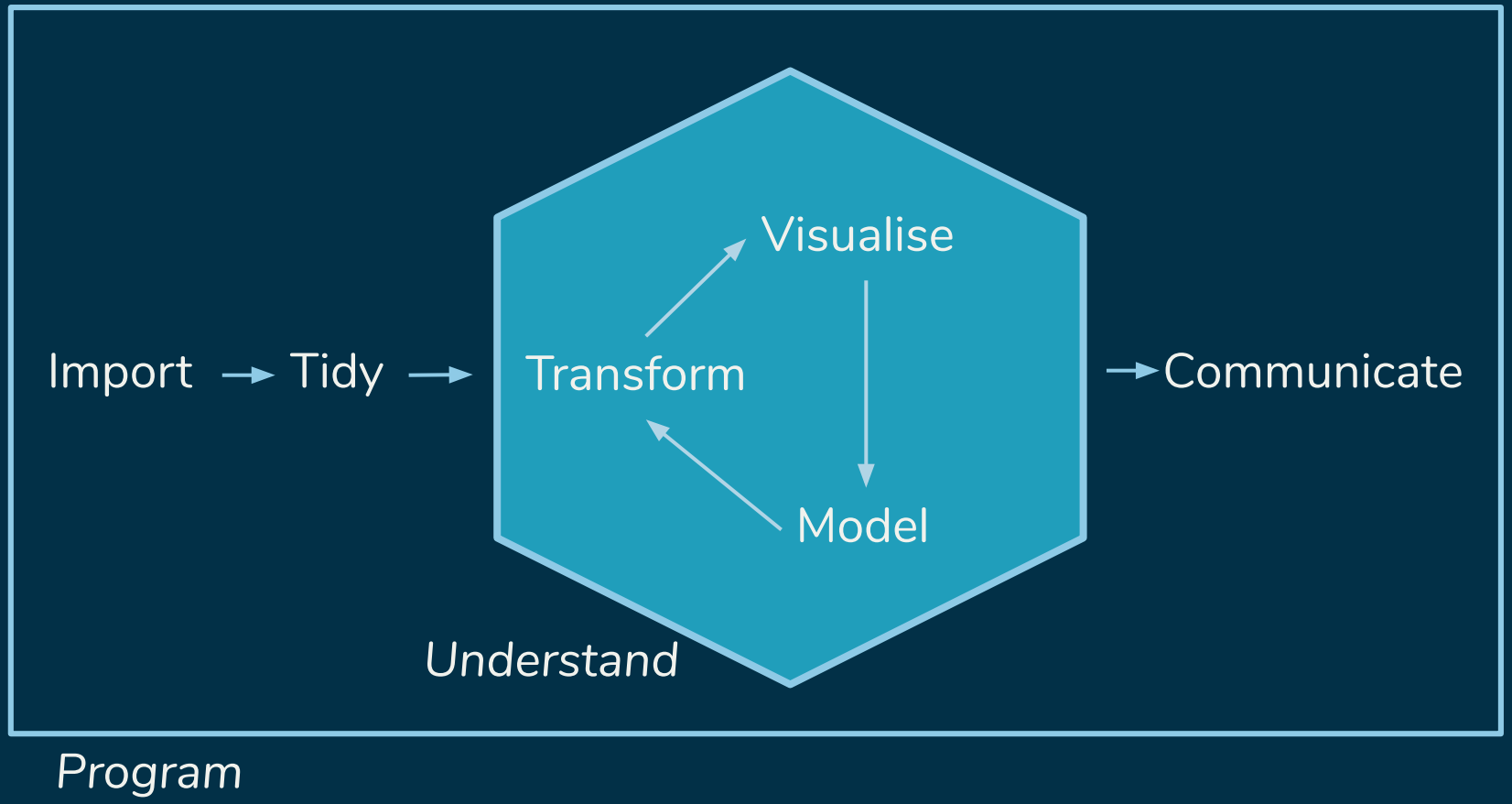
Meet the assistants

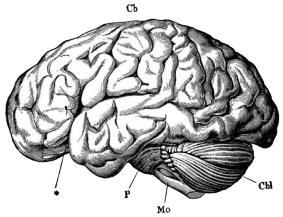


Naviz

William

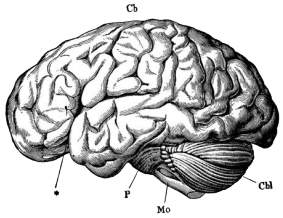
Data science is the art of turning raw data into understanding


Import
Tidy
Transform
Visualise
Model
Communicate
Understand
Program



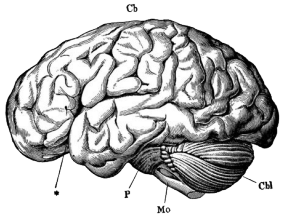
Human thought ⟷ Machine Language

Source: https://github.com/rstudio-education/arm-workshop-rsc2019



Human thought → Machine Language

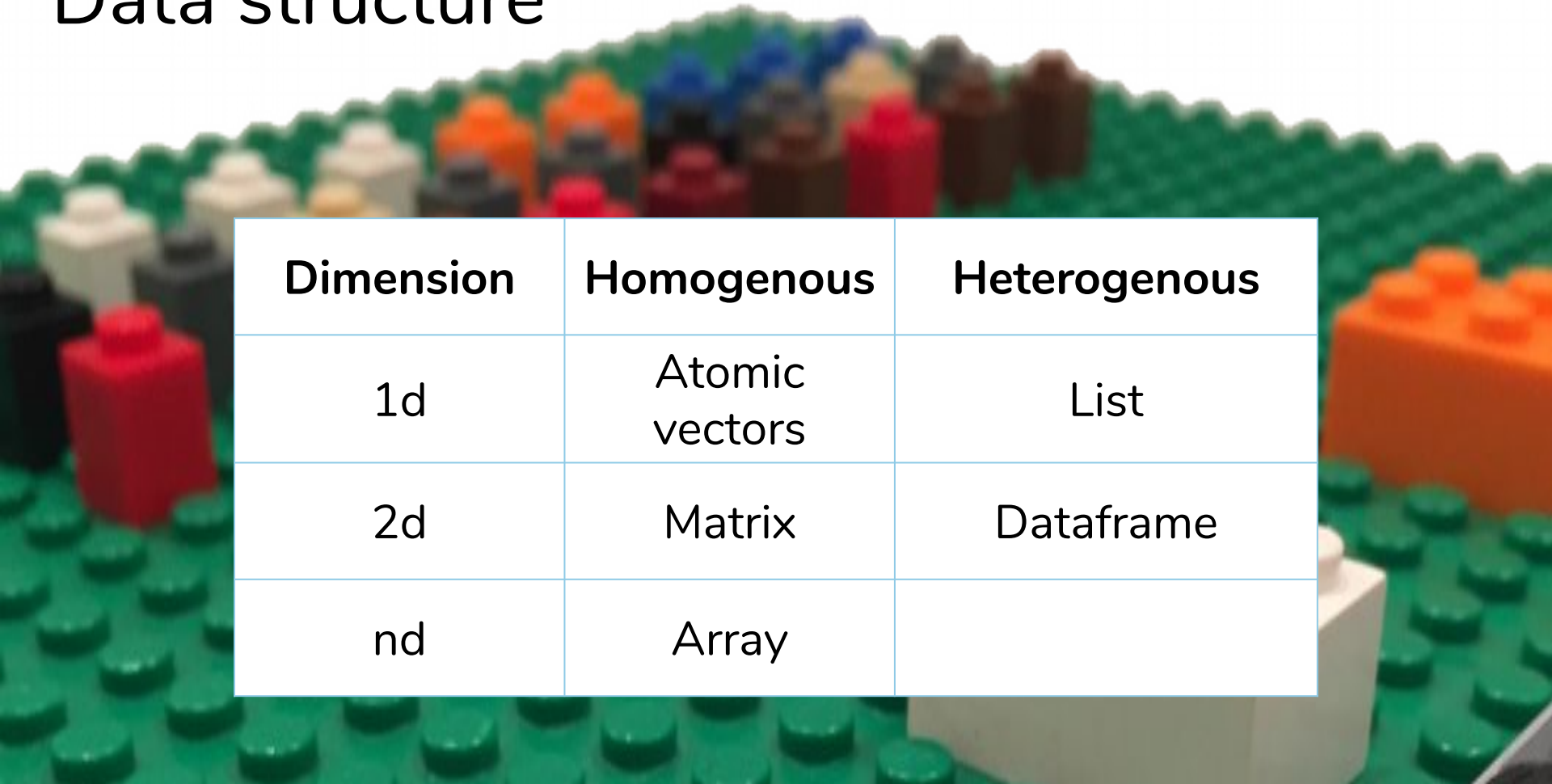
Source: https://github.com/rstudio-education/arm-workshop-rsc2019



Human thought → Machine Language

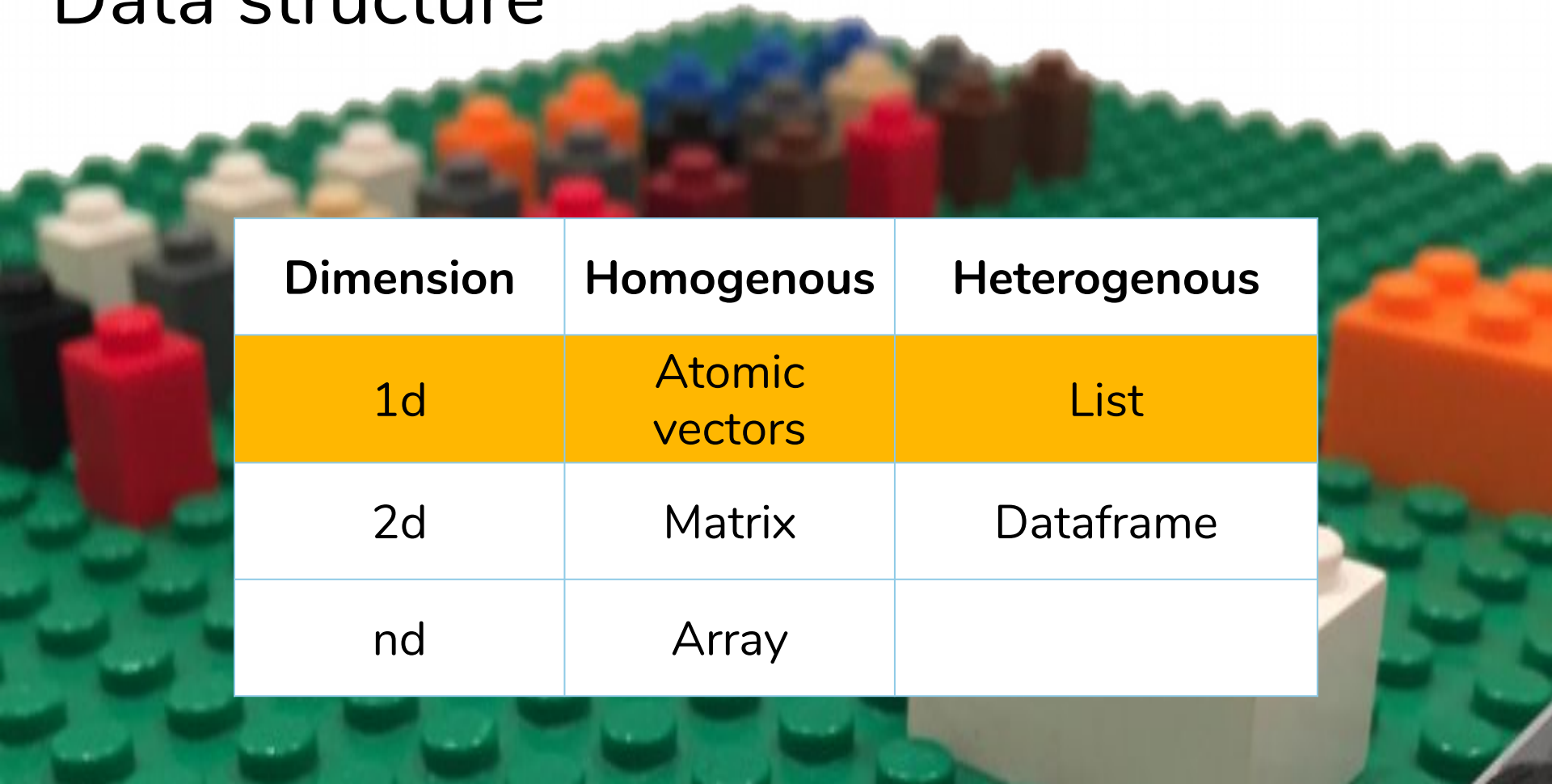
Source: https://github.com/rstudio-education/arm-workshop-rsc2019

Data structure

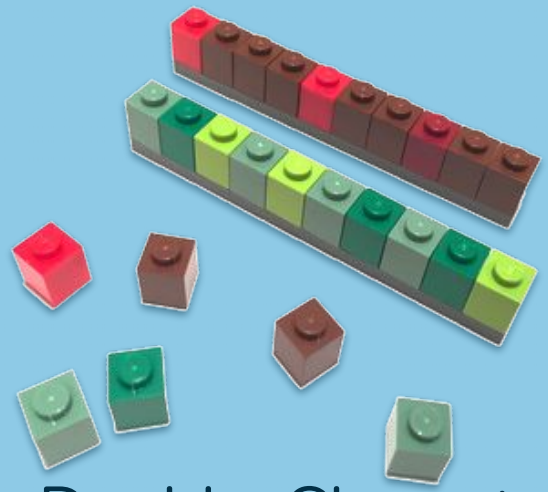


Dimension	Homogenous	Heterogenous
1d	Atomic vectors	List
2d	Matrix	Dataframe
nd	Array	

Data structure

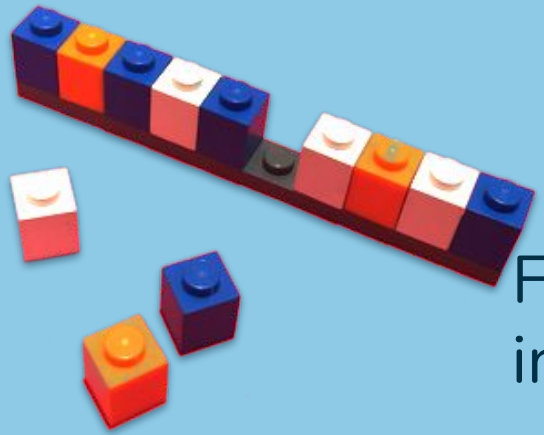


Dimension	Homogenous	Heterogenous
1d	Atomic vectors	List
2d	Matrix	Dataframe
nd	Array	



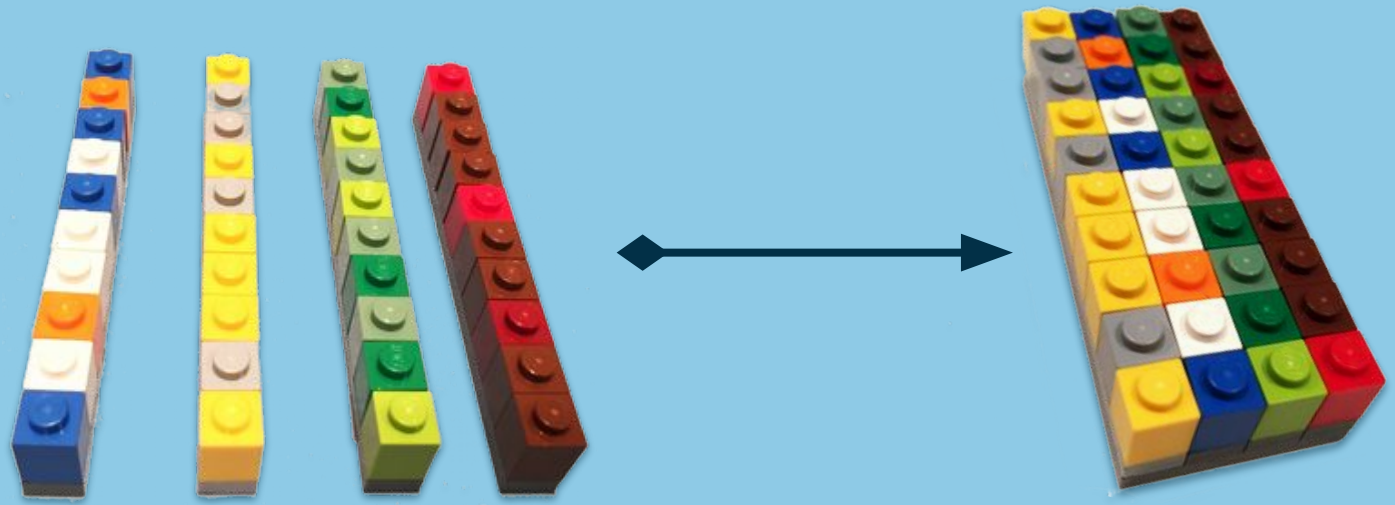
Integer, Double, Character

Logical



Factor (basically integer with class)

Artwork by @JennyBryan



Vectors of same length

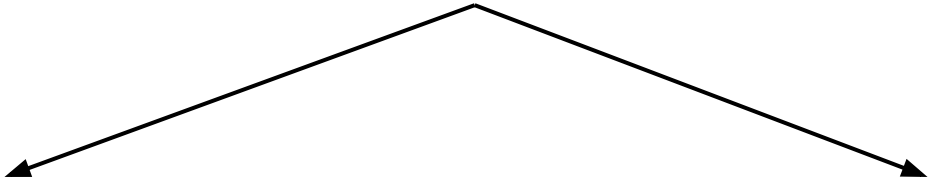
Dataframe

Artwork by @JennyBryan

How do we process

the data?

```
function(arg1, arg2, arg3, ...)
```



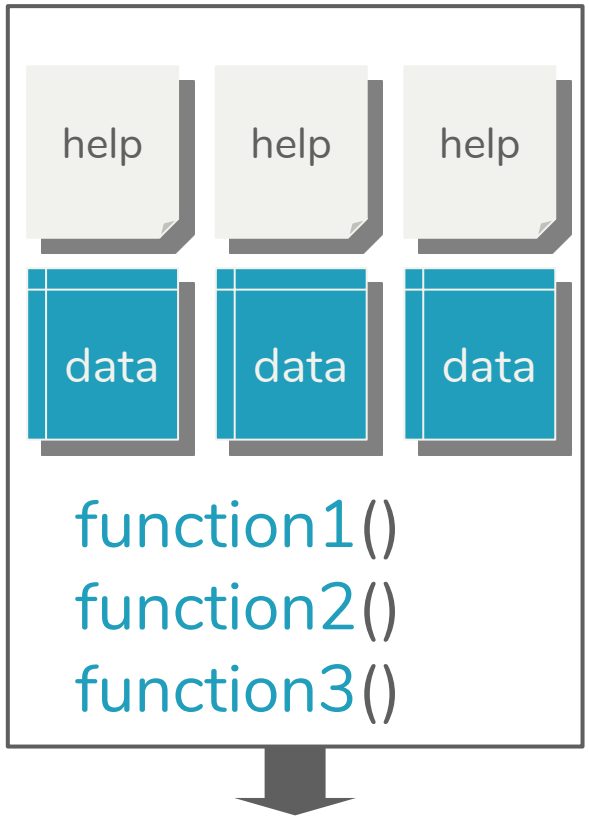
change_the_env(...)

calculate_value(...)

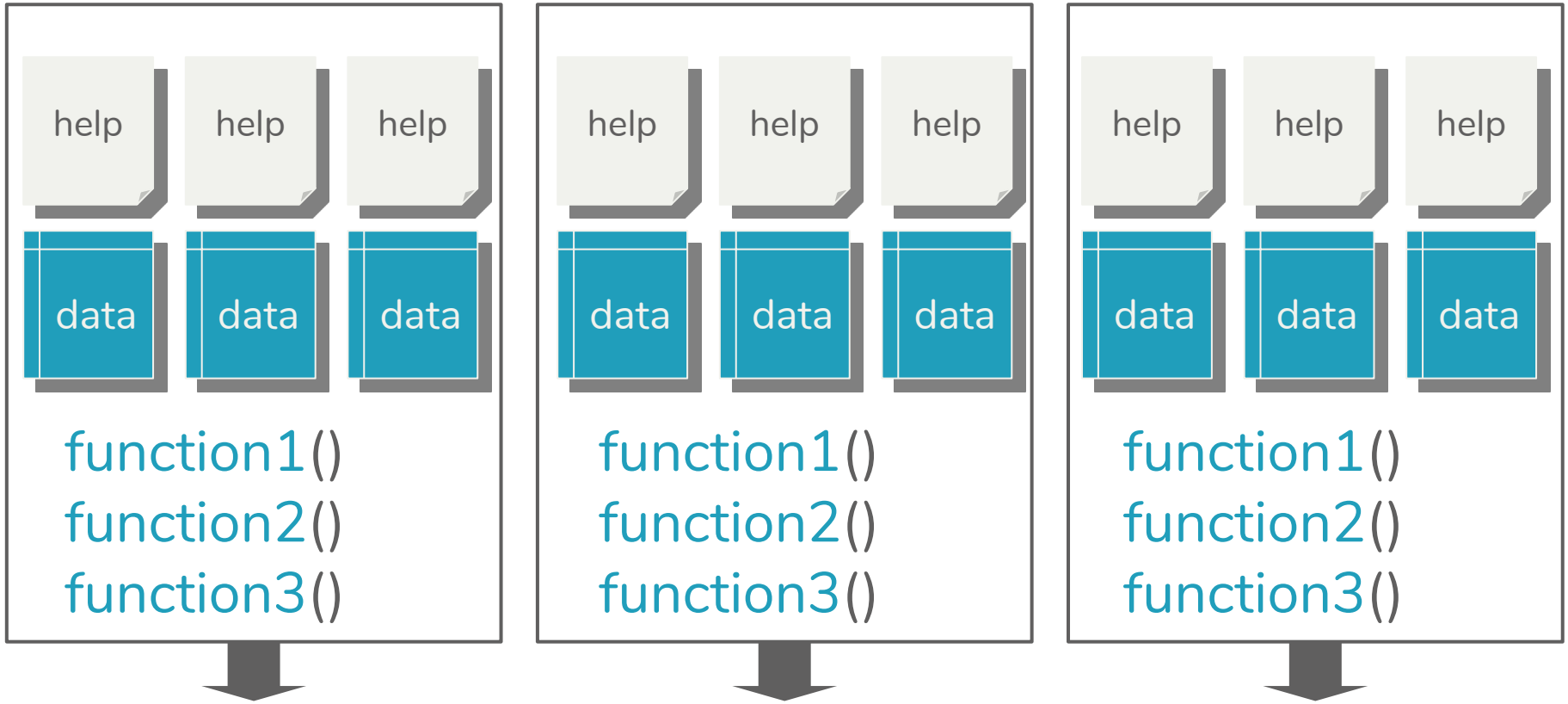
assign. <- , =, ->

- arguments are contexts of a function
- arguments are matched by name, or
- arguments are matched by position, **be careful!**

function1()
function2()
function3()



stats, graphics, grDevices, utils,
datasets, methods, base

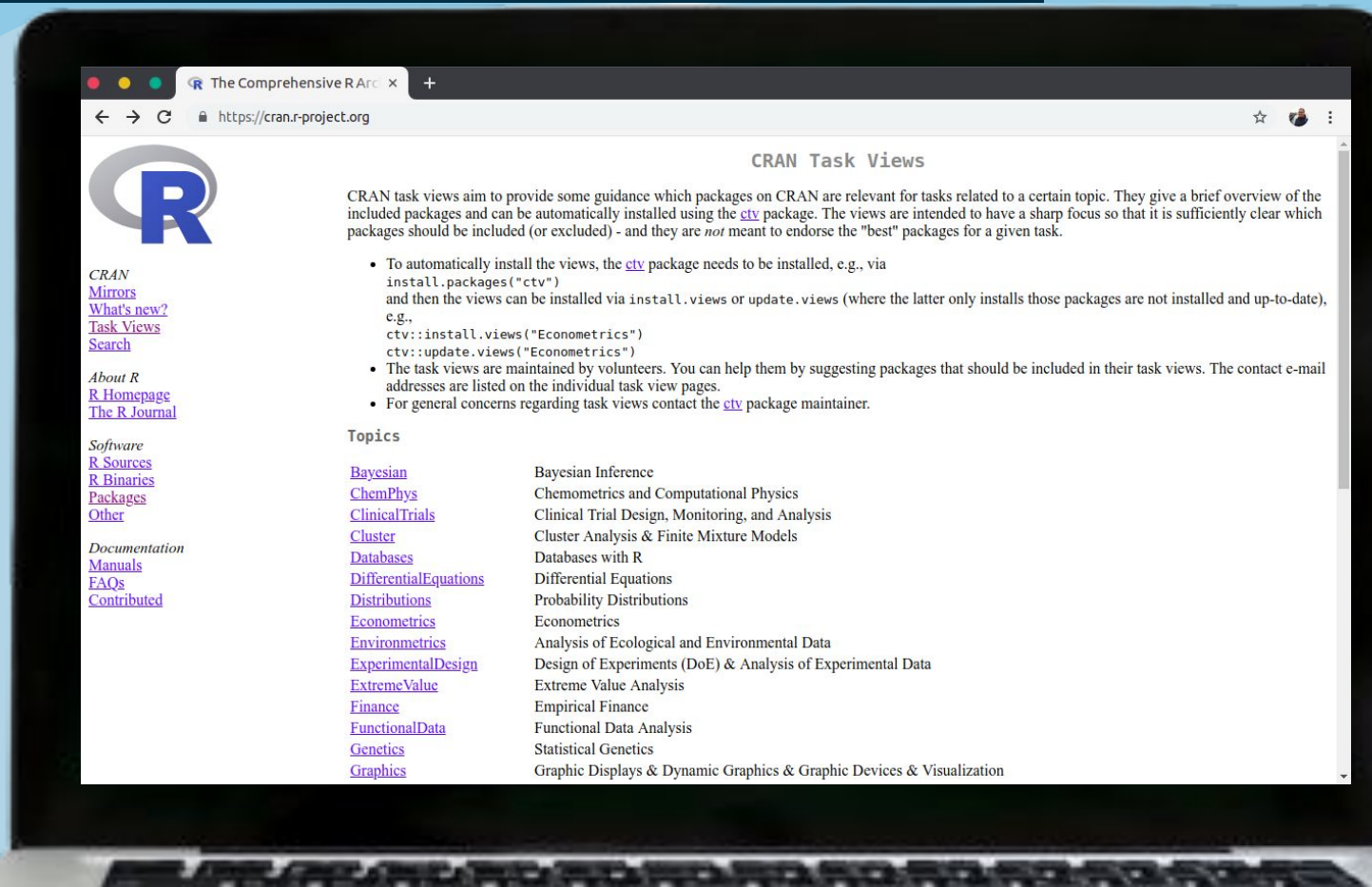


stats, graphics, grDevices, utils, datasets, methods, base

R packages

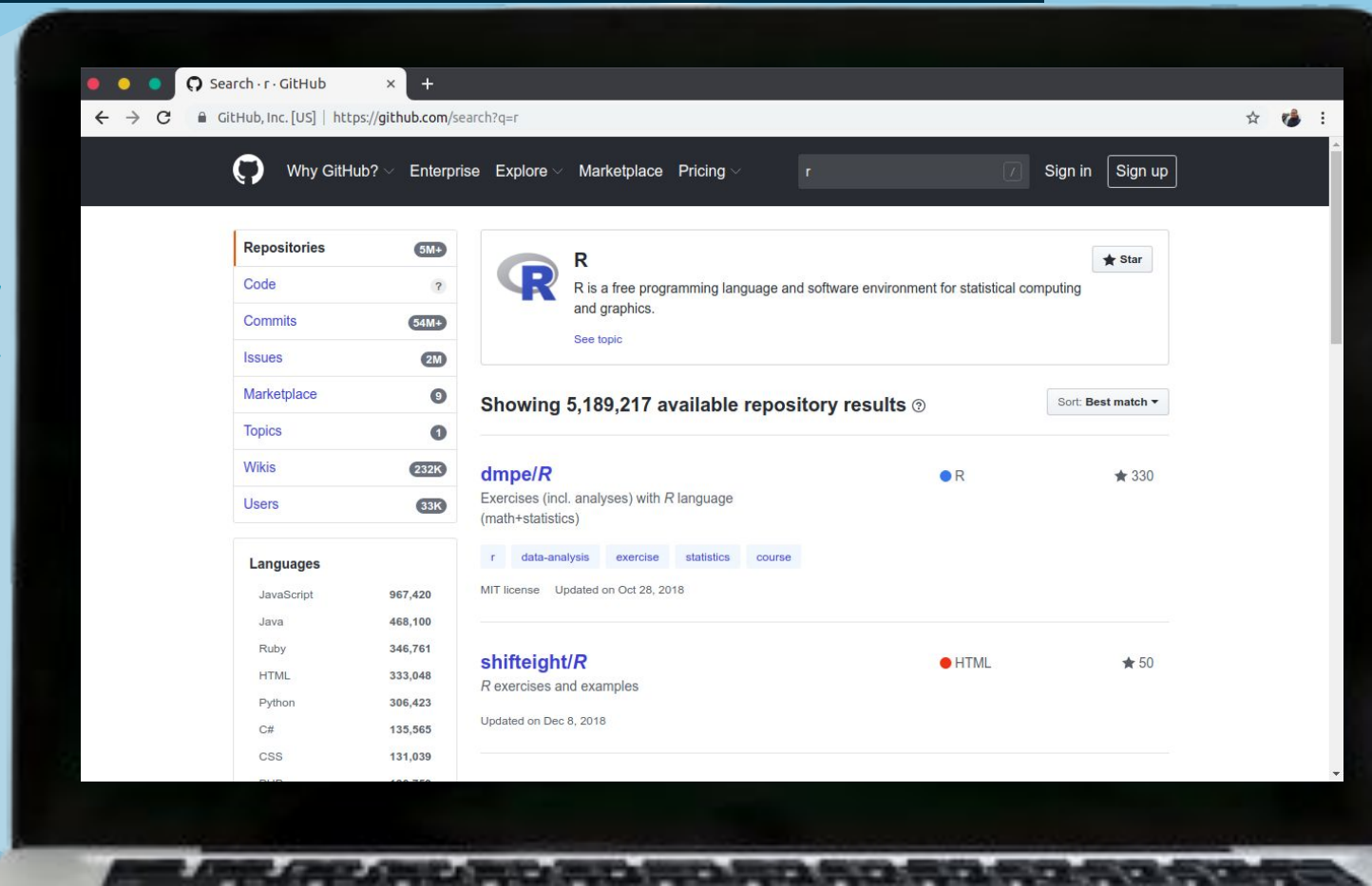
CRAN Task Views

https://cran.r-project.org/web/views/



GitHub

https://github.com/search?q=r



Installing package, only once

```
install.packages(“pkg”) # from CRAN/local
remotes::install_github(“user/pkg”) # from GitHub
remotes::install_bioc(“repo”) # from Bioconductor
```

Loading package, once in every session

```
library(pkg)
pacman::p_load(pkg) # load or install if not
available
```

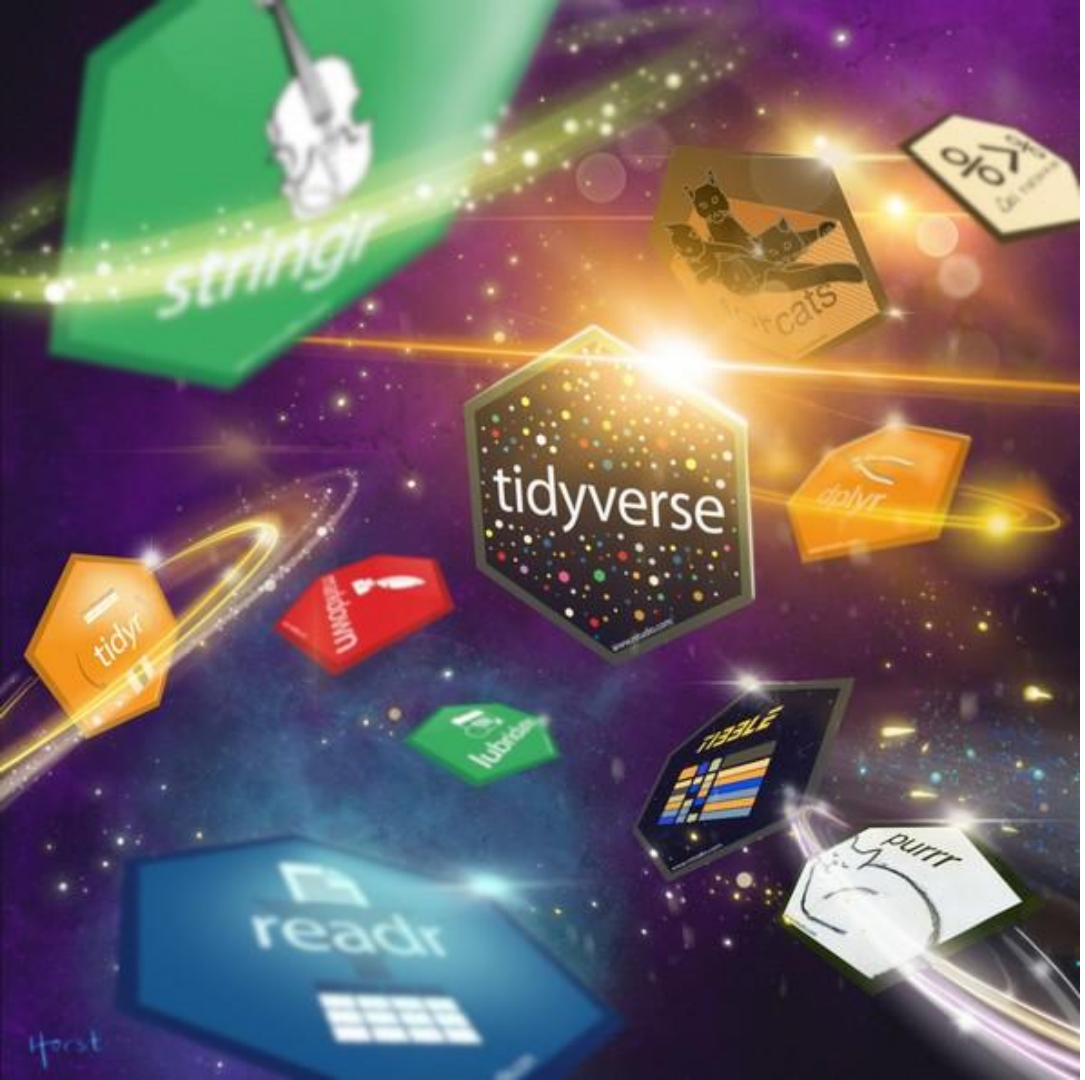
A lot of R packages

to use! :D

A lot of R packages

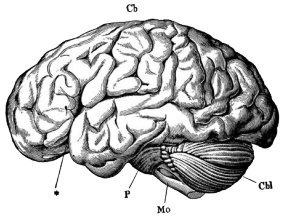
to use! :(


tidyverse
www.rstudio.com



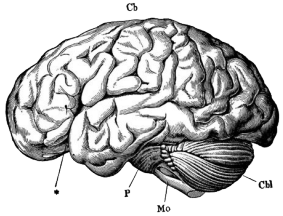
The tidyverse is an opinionated **collection of R packages** designed for **data science**. All packages **share** an underlying **design** philosophy, **grammar**, and **data structures**.

Artwork by @allison_horst



Human thought → Machine Language

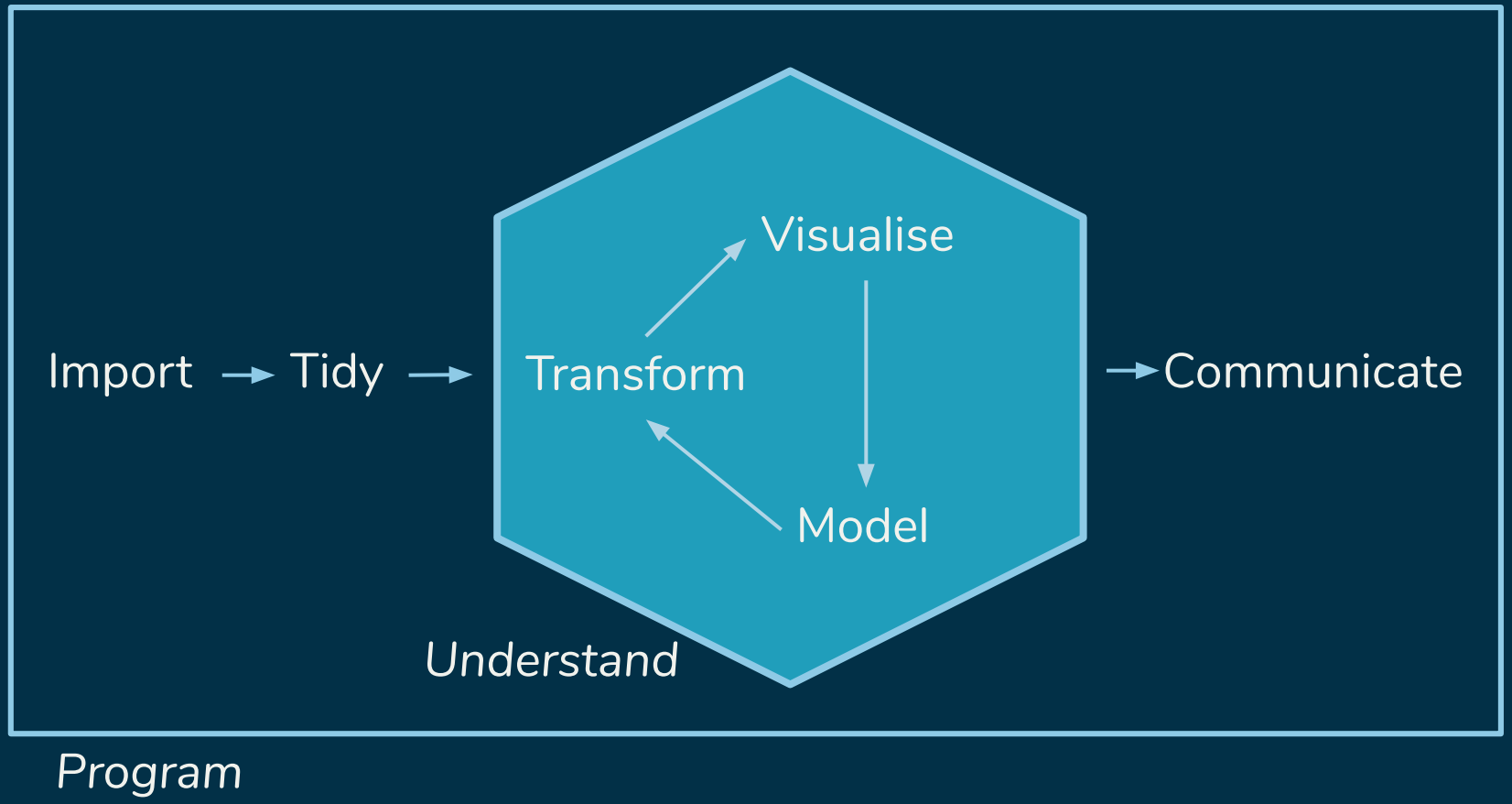
Source: https://github.com/rstudio-education/arm-workshop-rsc2019

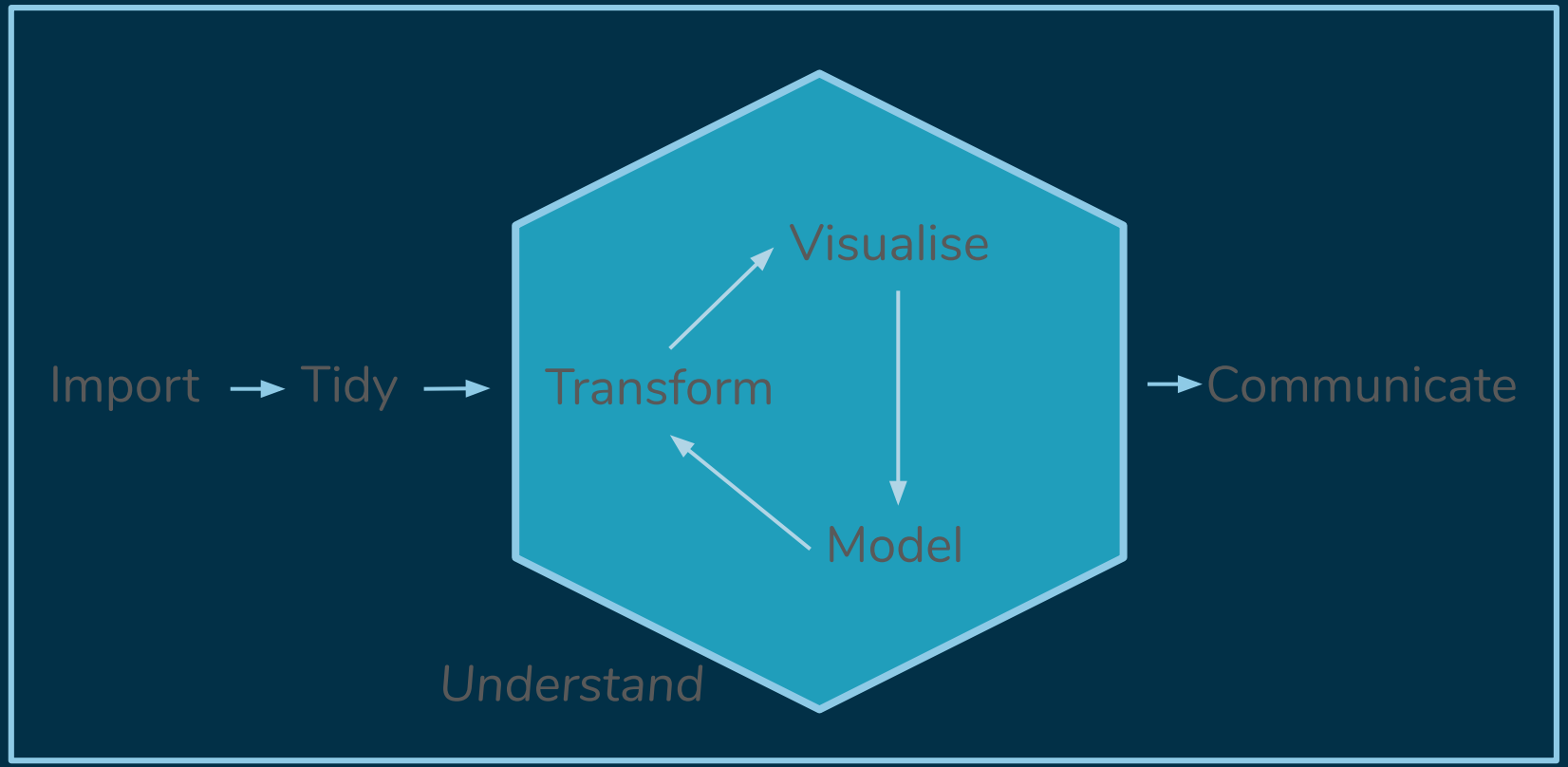


Human thought

Machine Language

Source: https://github.com/rstudio-education/arm-workshop-rsc2019


Import
Tidy
Transform
Visualise
Model
Communicate
Understand
Program


Import
Tidy
Transform
Visualise
Model
Understand
Communicate
Program


tidyverse
www.rstudio.com

tidycode

tidymodels

tidync

tidyverts

tidycensus

tidygraph

tidytext

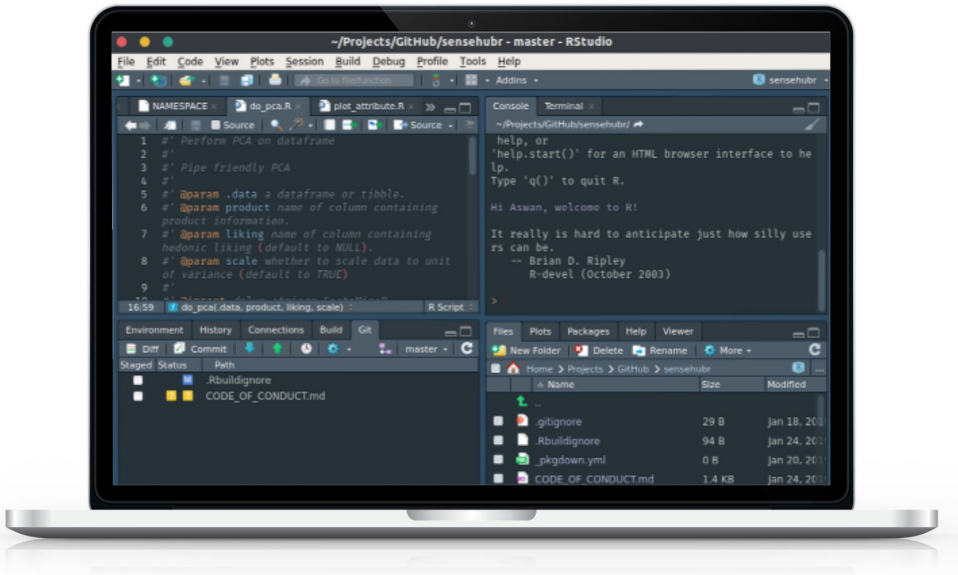
tidygenomics

Where to code?

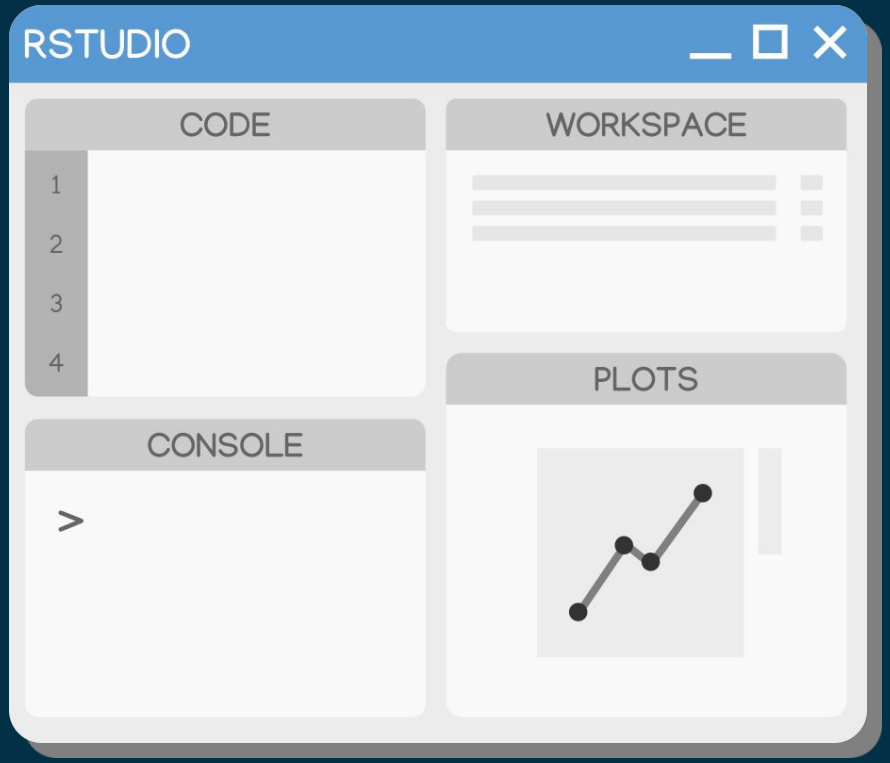
RStudio®

Main features:

- Console
- Syntax-highlighting editor
- Tools for plotting, history, debugging and workspace management



rstudio.com/products/rstudio/download/



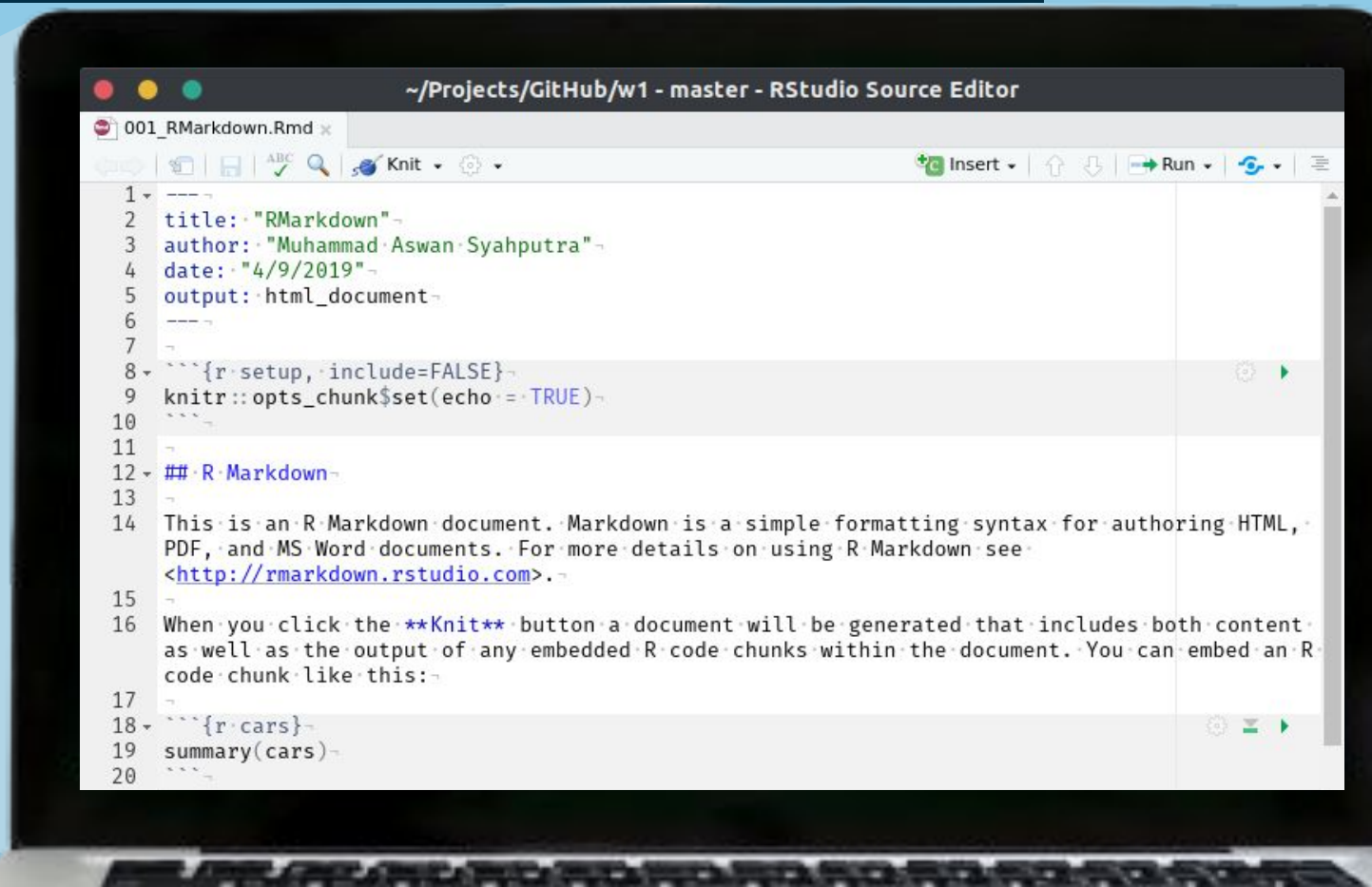
- **Tab**, autocompletion & path navigation
- **Alt + -**, for assignment operator <-
- **Ctrl + Shift + M**, for pipe operator %>%
- **Ctrl + Enter**, run current line code/example on help page
- **Ctrl + Up**, search for code history on console or editor pane
- **Alt + Up/Down**, move code to above or below
- **Alt + Shift + Up/Down**, copy code to above or below
- **Ctrl + D**, delete current line
- **Ctrl + Shift + F10**, restart R session
- **Ctrl + Alt + B**, run code up to current line

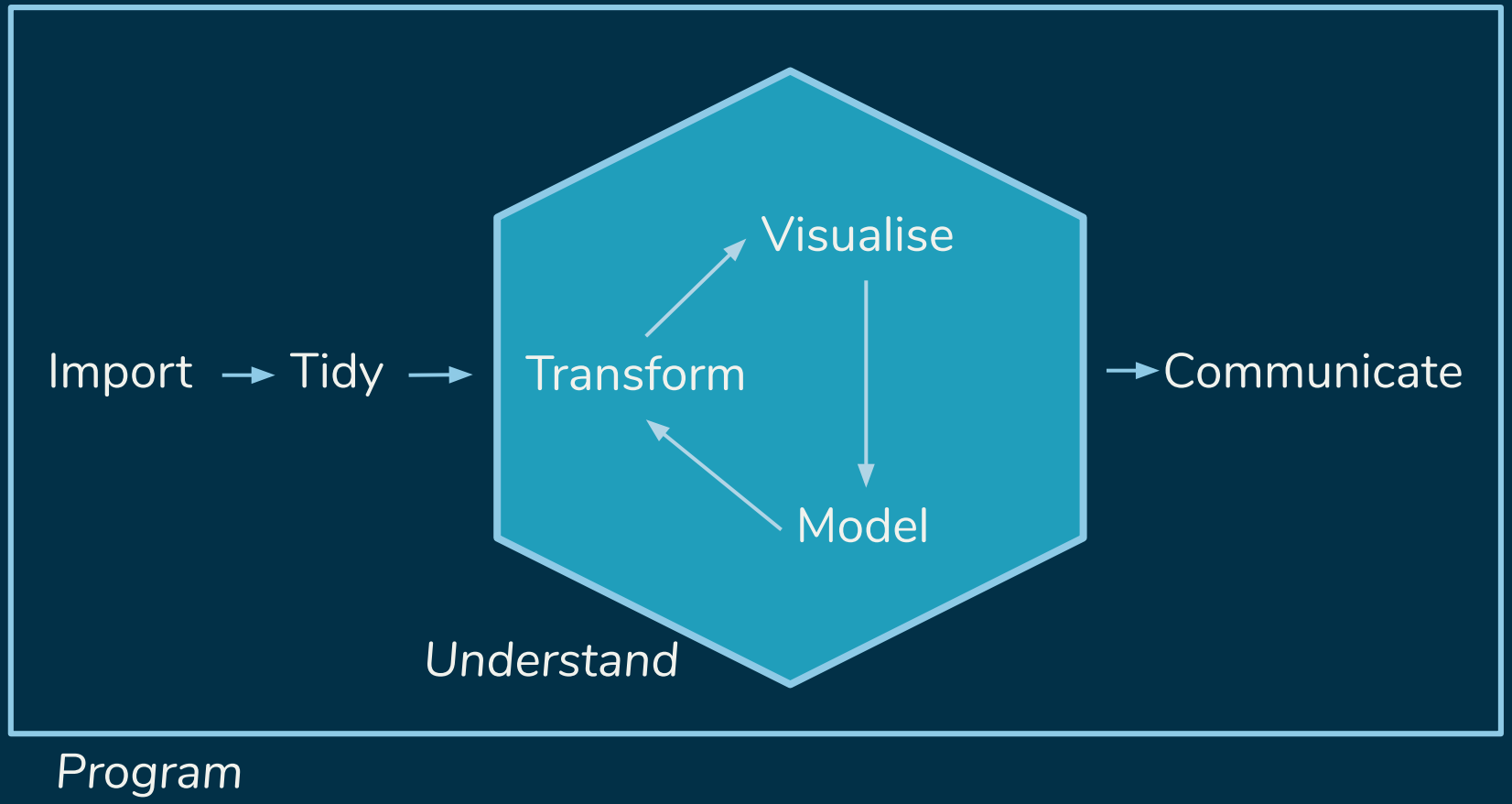
Cheatsheet: https://github.com/rstudio/cheatsheets/raw/master/rstudio-ide.pdf

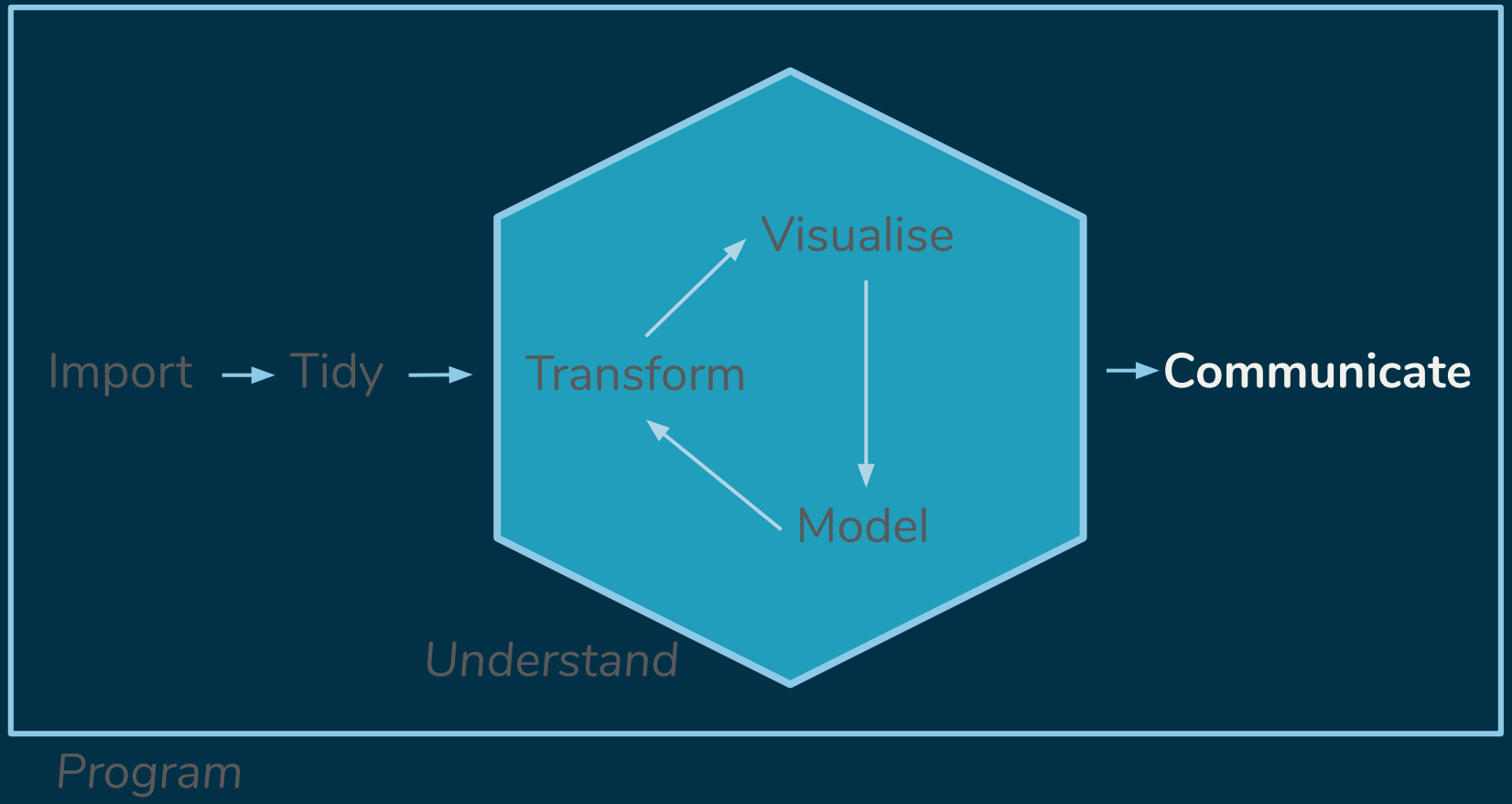
Artwork by @allison_horst

R Markdown

A document format for authoring data science project




Import
Tidy
Transform
Visualise
Model
Communicate
Understand
Program


Import
Tidy
Transform
Visualise
Model
Communicate
Understand
Program

- Use script (R Markdown or R Script), try to avoid console
- Use Projects, not setwd(...) in script
- Set stringsAsFactor = FALSE, but not in the .Rprofile
- Ctrl+Shift+F10 and Ctrl+Alt+B to clean up,
- not rm(list=ls())
- Learn the handy shortcuts
- Do not save and load .Rdata
- Use version control system: git!

Reading: happygitwithr.com

Download: git-scm.com

With great codes,
comes great bugs!

- (not) Uncle Ben

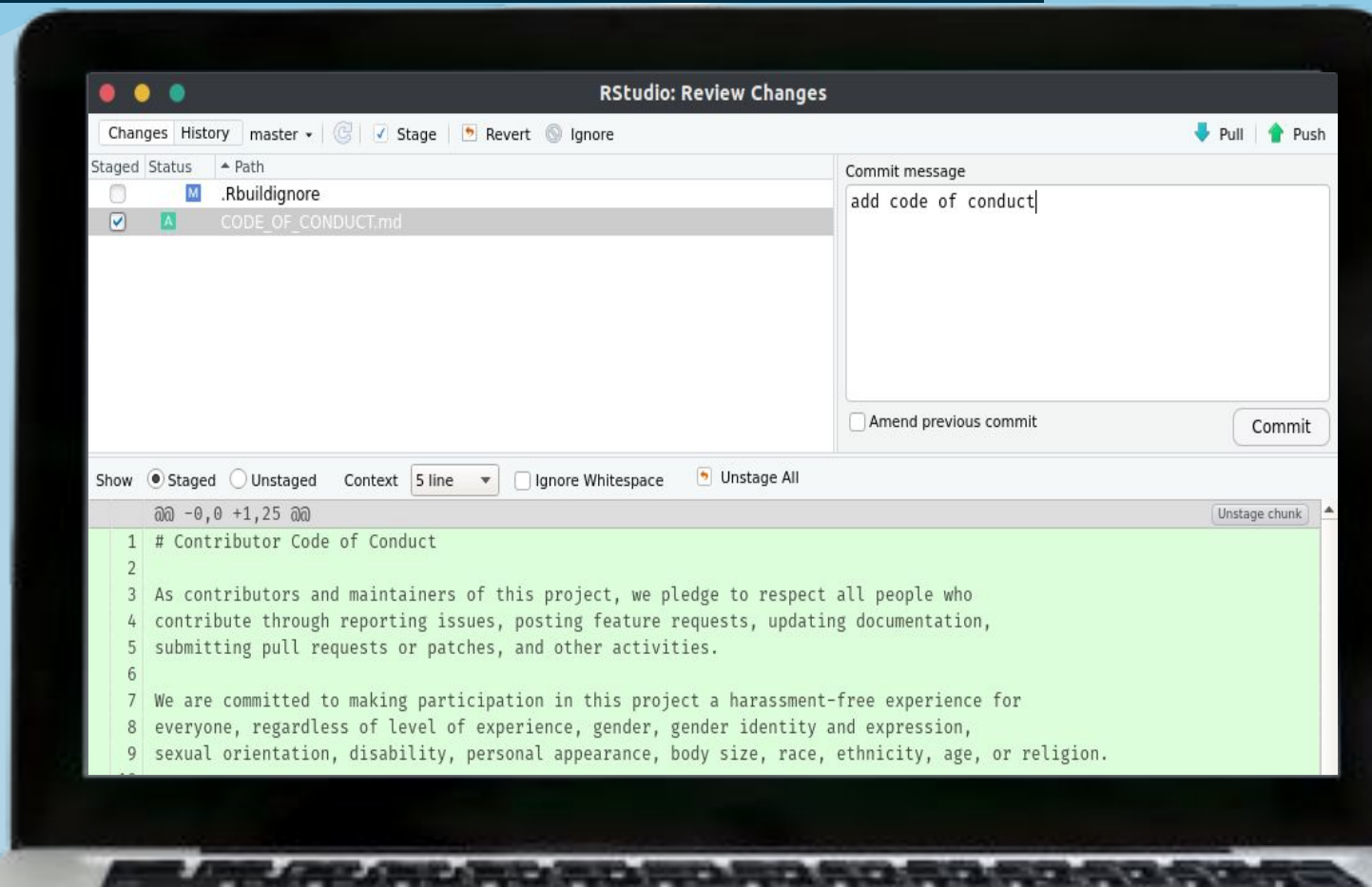
Store and share! Why sharing your work? Motivation here.

- git clone https://github/user/repo
- Do some works!
- git add file.R or git add .
- git commit -m “what you have done”
- git push origin master
- Repeat: work, git add, git commit, git push

git

- git init
- git remote add origin https://github/user/repo
- Do some works!
- git add file.R or git add .
- git commit -m “what you have done”
- git push -u origin master #use -u only once
- Repeat: work, git add, git commit, git push

It is available in RStudio!



Let's get them!

- Download R at **cran.r-project.org**
- Download RStudio at **rstudio.com**
- Download git **git-scm.com**

How to **set**?

- Create account at **github.com**
- In Rstudio, Tools – Global Options – Git/SVN – Browse git executable:
  - Windows:
  - **C:/Program Files/Git/bin/git.exe**
  - UNIX/UNIX-like:
  - **/usr/bin/git**

How to set?

- Open git bash/shell/terminal
- Run:

```
git config --global user.email
“email@domain.com”
```

```
git config --global user.name
“Your Name”
```

Let's get **started**!

- Go to **github.com/r-academy/tidyds**
- Hit 'Fork' button
- Click 'Clone or Download', copy the URL
- In RStudio, File – New Project – Version Control – Git. Paste URL
- In File pane (bottom-right), click vignettes-'m1_essentials.Rmd' to open it

Working directory

```
> fs::dir_tree()
.
├── 003_kamisdata_Debat-Pilpres1-2019.Rproj
├── Dockerfile
├── R
│   ├── cari.R
│   └── impor.R
├── README.md
├── data
│   └── debat-pilpres1-2019.rda
├── data-raw
│   └── debat_pilpres1_2019.R
├── install.R
└── vignettes
    ├── aswansyahputra-frekuensidansentimen.Rmd
    ├── aswansyahputra-frekuensidansentimen.html
    └── aswansyahputra-frekuensidansentimen_files
> 
```

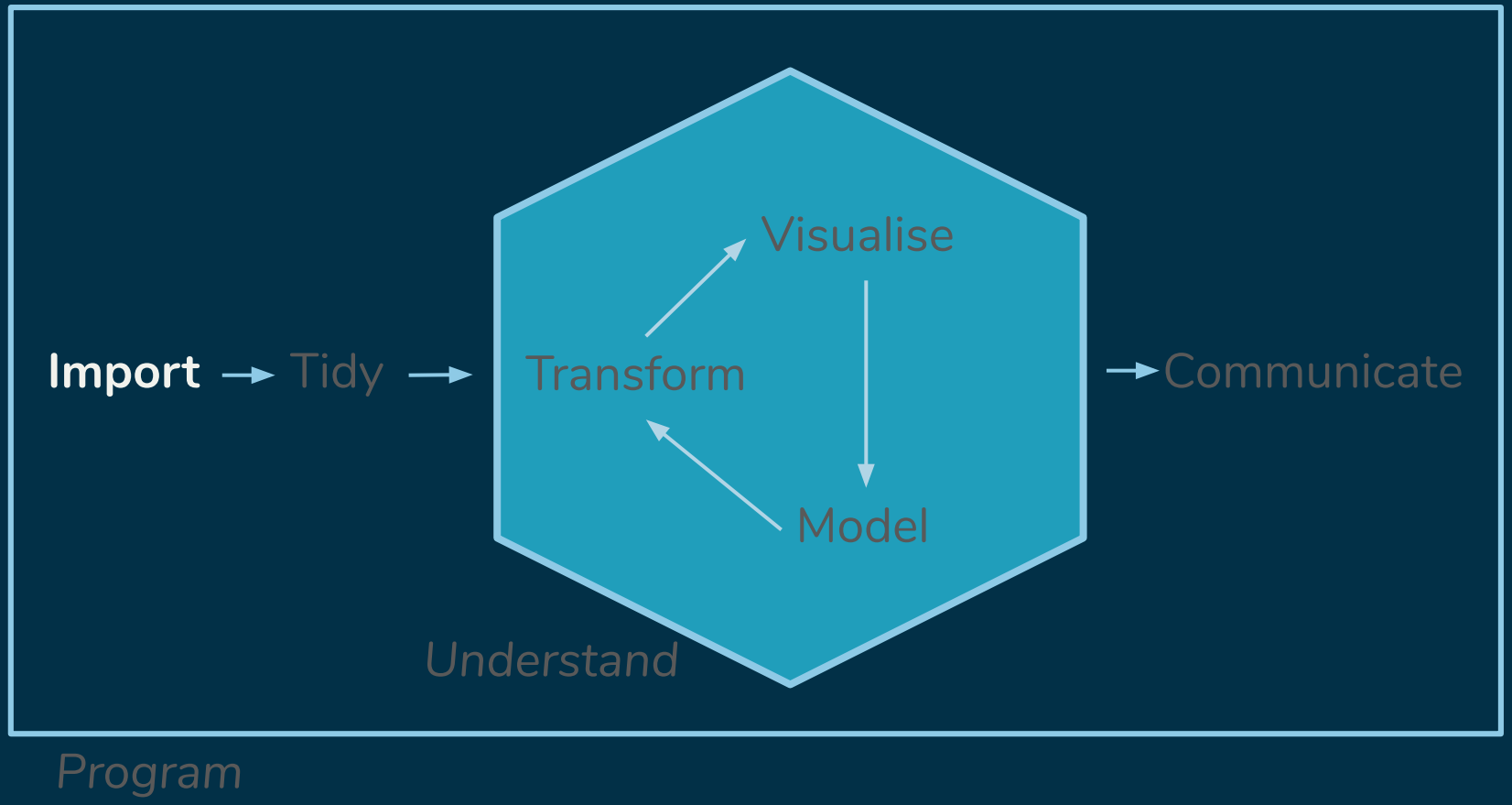
3 principles for naming files:

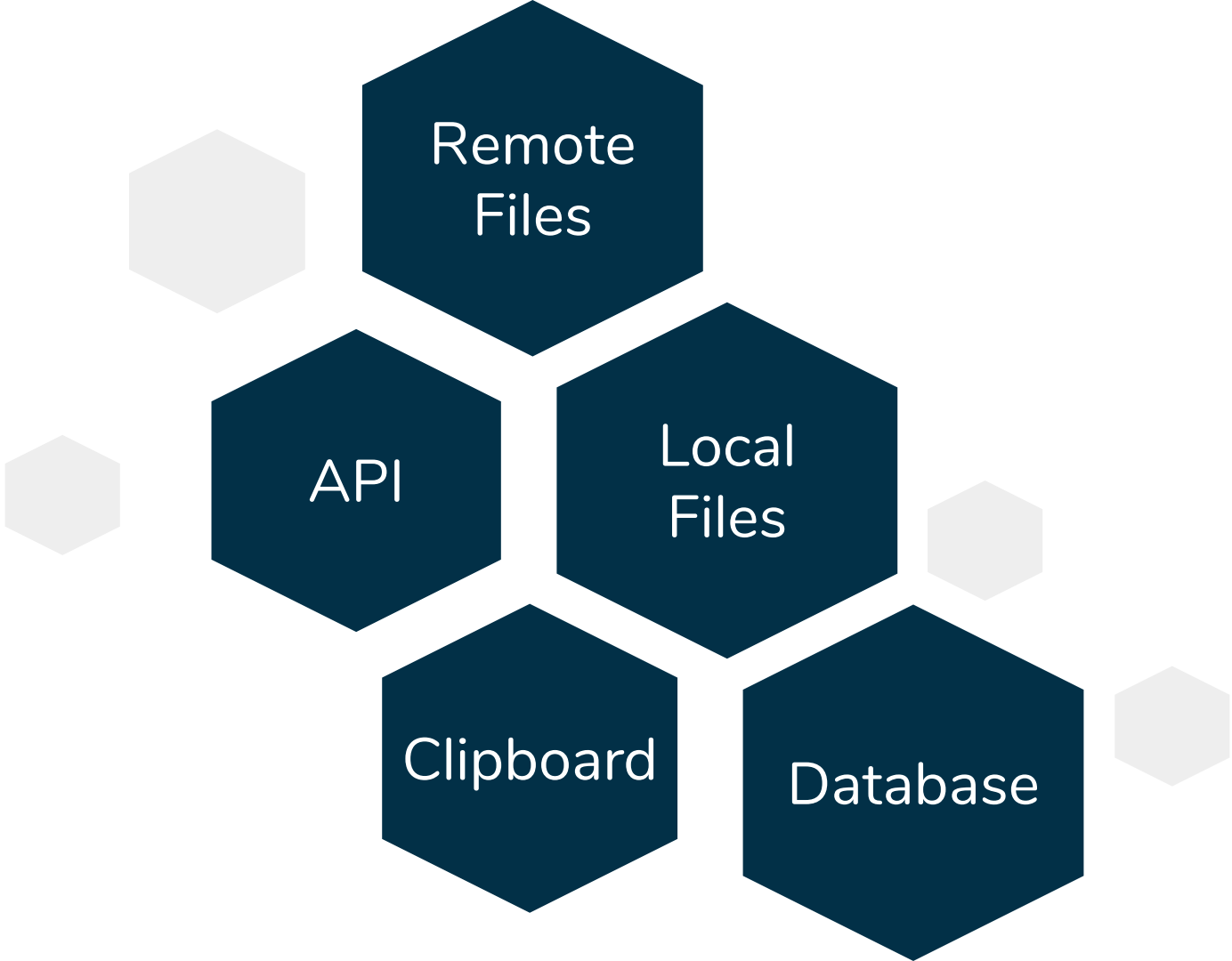
- Machine readable
- Human readable
- Default ordering

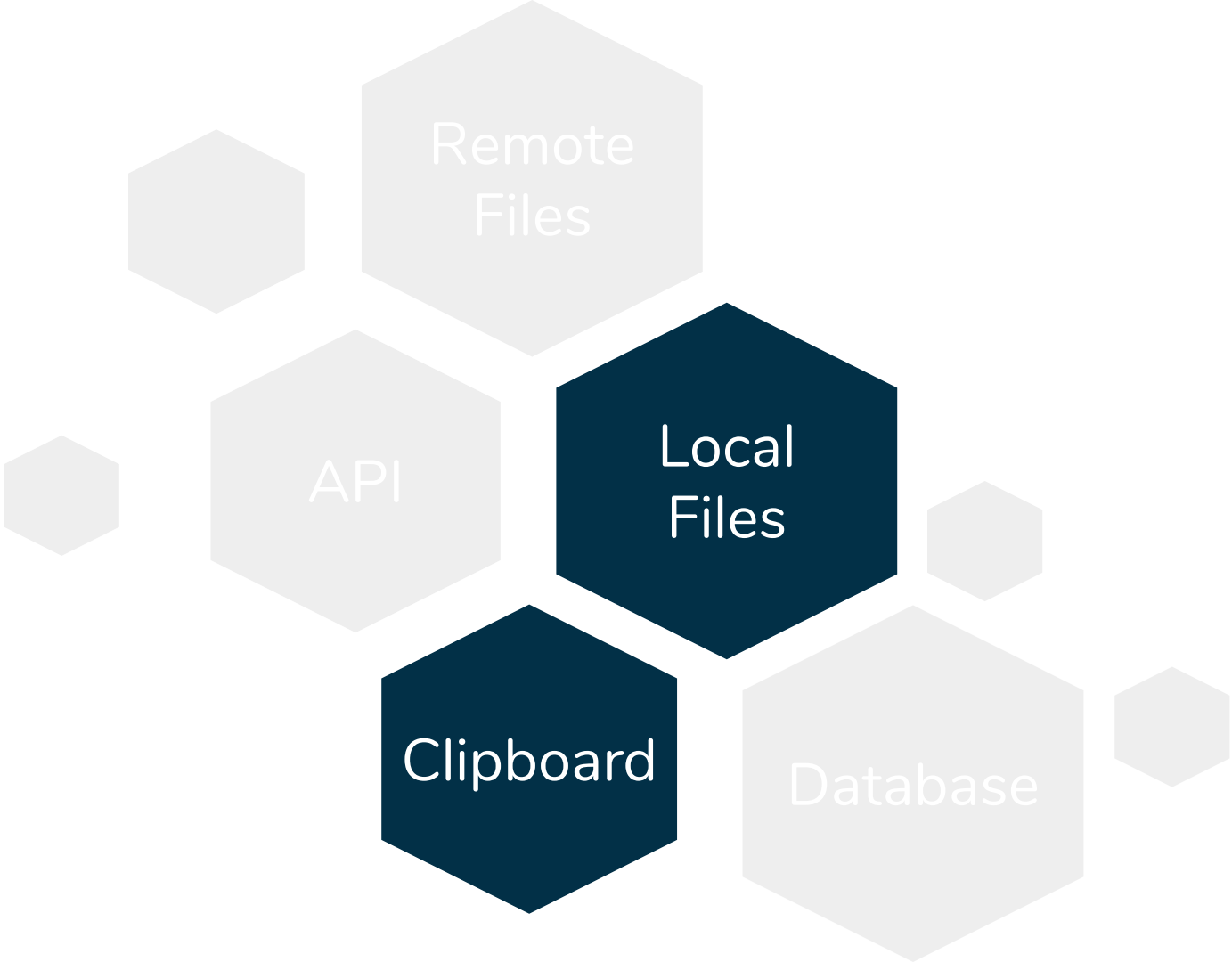
More info:
speakerdeck.com/jennybc/how-to-name-files

How to use **git**?

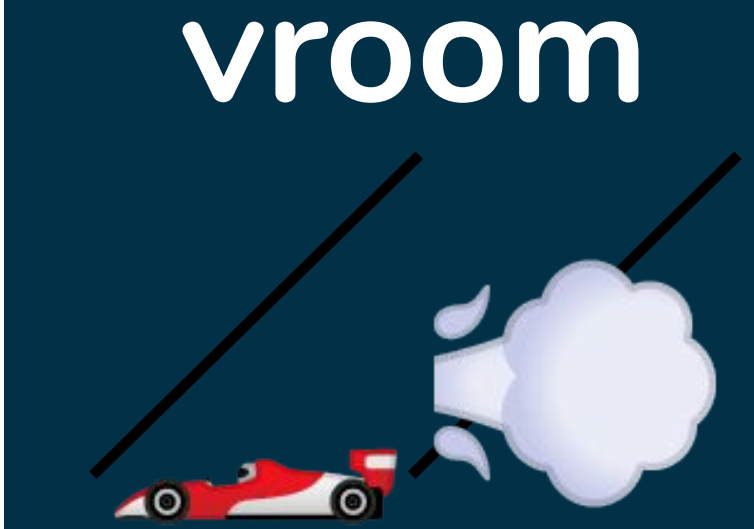
- In Environment Pane, hit ‘Git’ tab
- Click commit, a window pane will appear
- Select all files (Ctrl + A), click ‘Stage’
- Fill commit message, the click ‘Commit’
- Hit ‘Push’ Button, done!
- You may check you GitHub now!


Import
Tidy
Transform
Visualise
Model
Communicate
Understand
Program


Remote Files
API
Local Files
Clipboard
Database


Remote Files
API
Local Files
Clipboard
Database

- vroom(): comma separated (CSV) files
- vroom(): tab separated files
- vroom(): general delimited files
- vroom(): fixed width files
- vroom(): tabular files where columns are separated by white-space.



Tame data is lovely

- Use syntactical names for column names
- Use consistent case: snake_case, camelCase, caterpillar.case
- Cast column type accordingly: <chr>, <dbl>, <lgl>, <date>, etc
- Treat <fct> carefully!
- Preferably turn implicit missing observation into explicit NA value

```
x <- something
for (i in seq_along(x) {
  function(x[[i]])
}
```

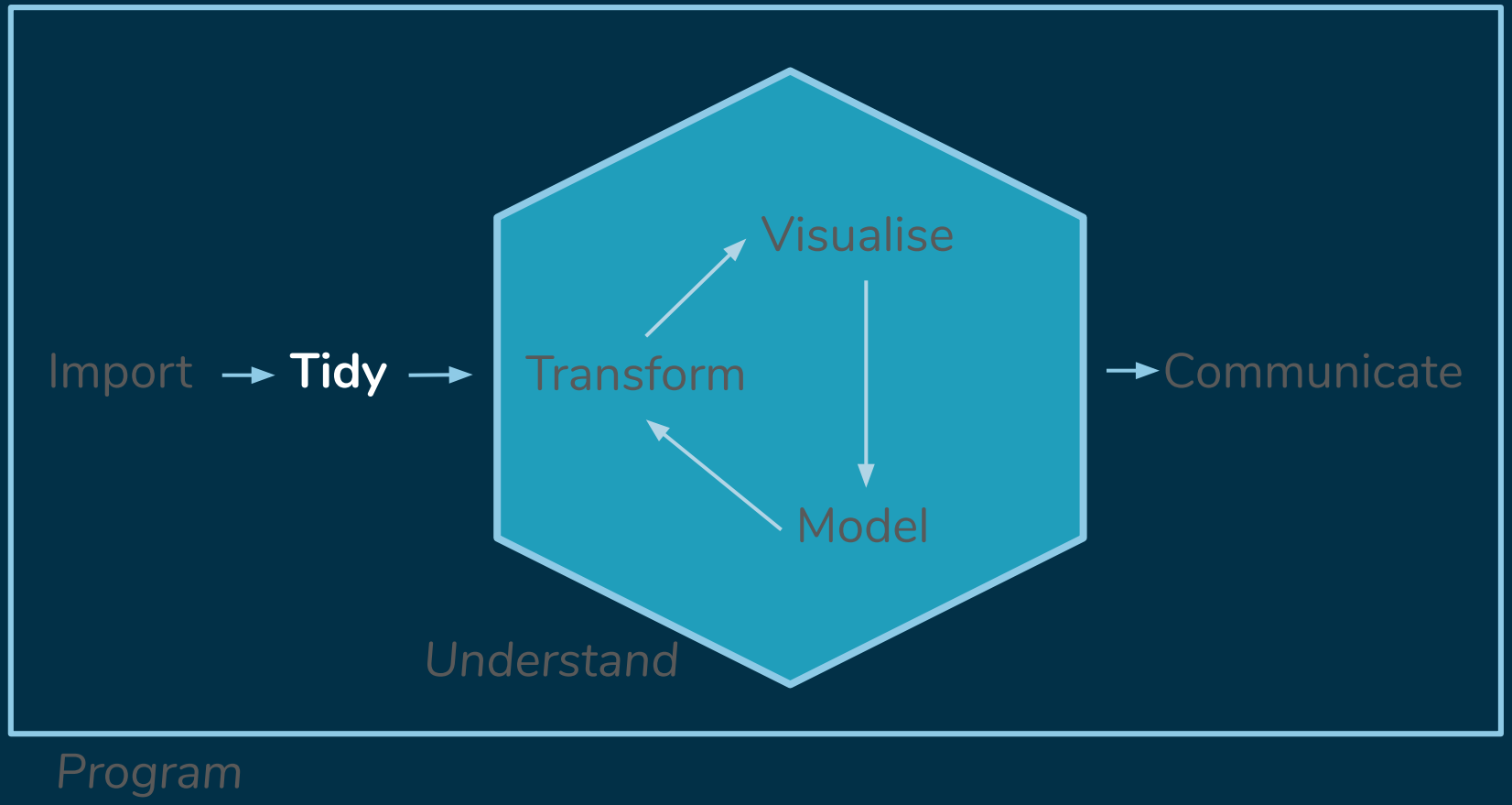
```
x <- something
*apply(x, function)
```

```
x <- something
map_*(x, function)
```

purrr

Let's do **practice**!

- Open 'm2_data-import.Rmd'
- Do not forget to push your works into GitHub!


Import
Tidy
Transform
Visualise
Model
Communicate
Understand
Program

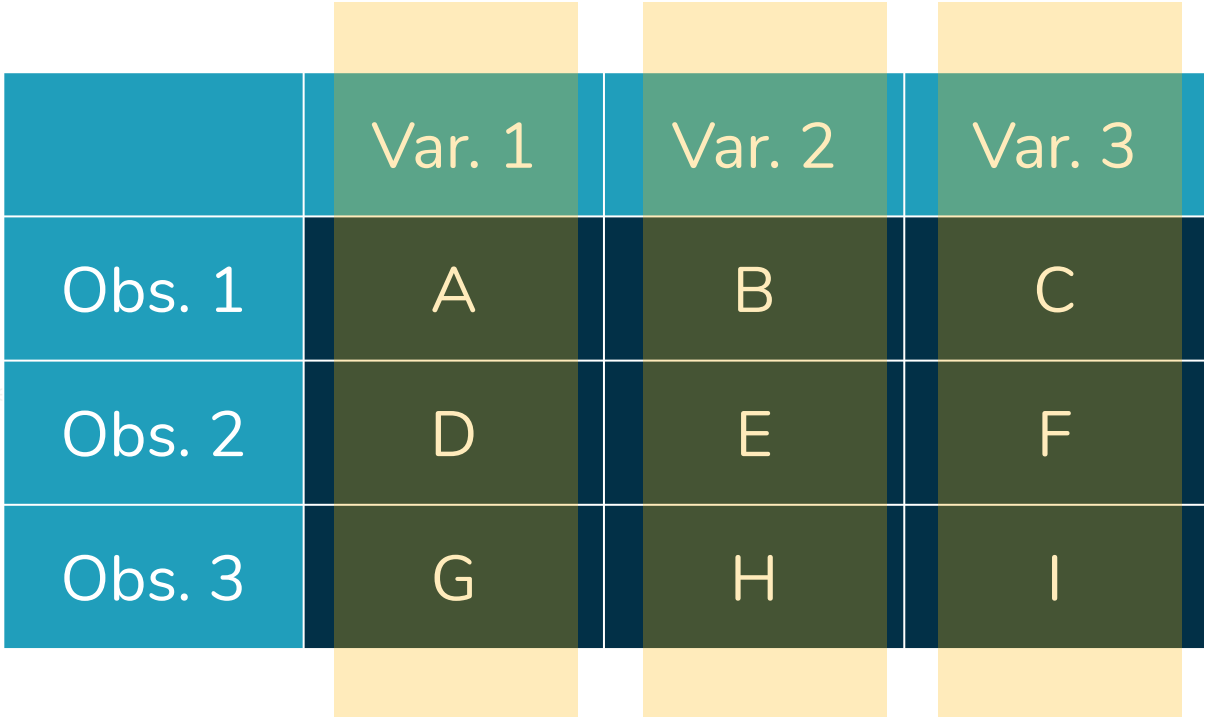
Tidy datasets are all alike,
but every messy dataset is
messy in its own way!

- *Hadley Wickham*

A Tidy dataset

	Name	Gender	Age
1	Jane	Male	44
2	Lisbon	Female	38
3	John	Male	NA

A variable has its own column



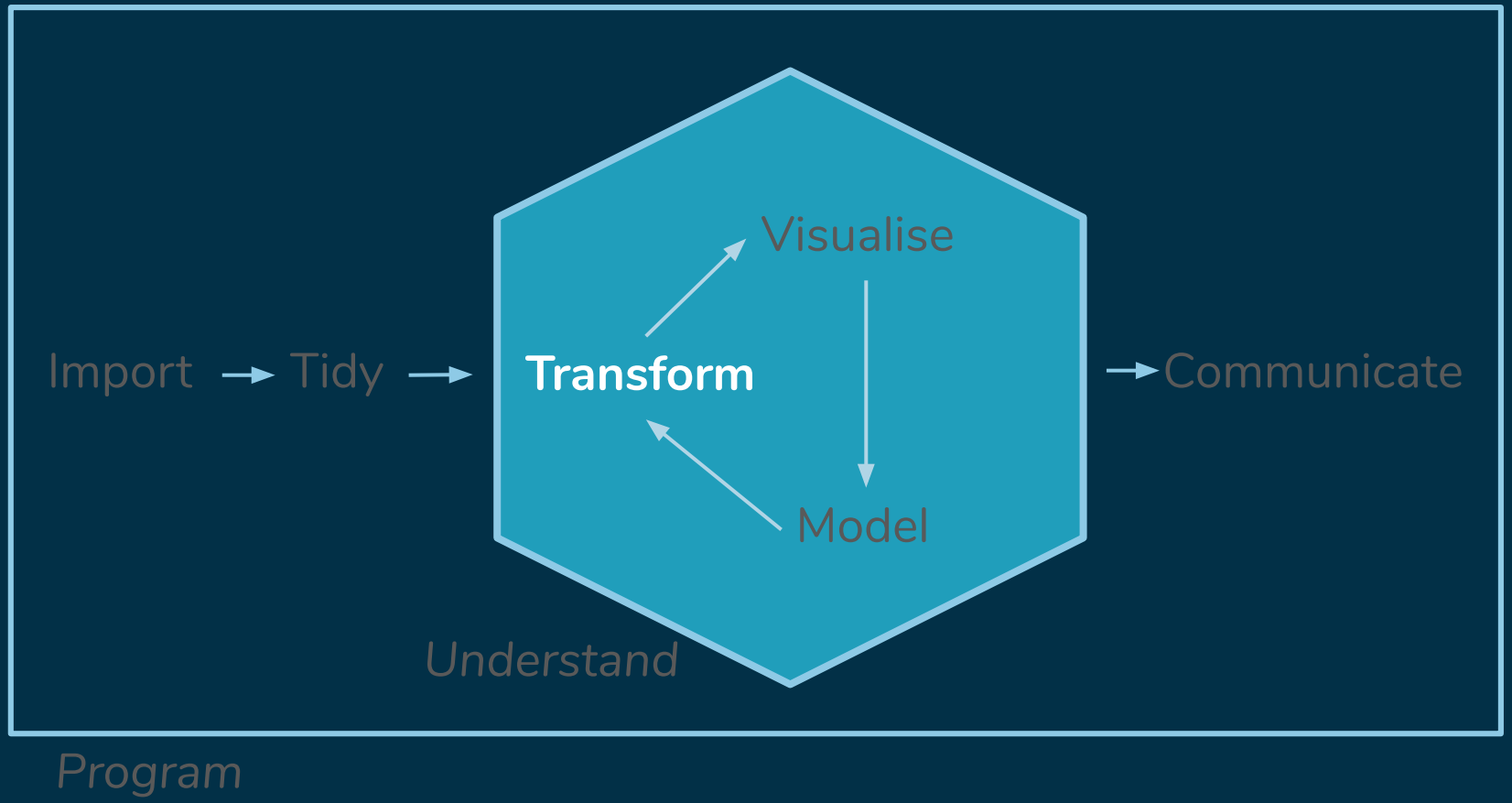
An observation has its own row

	Var. 1	Var. 2	Var. 3
Obs. 1	A	B	C
Obs. 2	D	E	F
Obs. 3	G	H	I

A value has its own cell

	Var. 1	Var. 2	Var. 3
Obs. 1	A	B	C
Obs. 2	D	E	F
Obs. 3	G	H	I

Artwork by @allison_horst


Import
Tidy
Transform
Visualise
Model
Communicate
Understand
Program

Artwork by @allison_horst

dplyr basic functions:

- filter() selects rows based on their values
- mutate() creates new variables
- select() picks columns by name
- summarise() calculates summary statistics
- arrange() sorts the rows

tidyr basic functions:

- gather() wide-format >> long-format
- spread() long-format >> wide-format
- fill() fills value based on previous entry
- complete() turns implicit missing values into explicit

Operators:

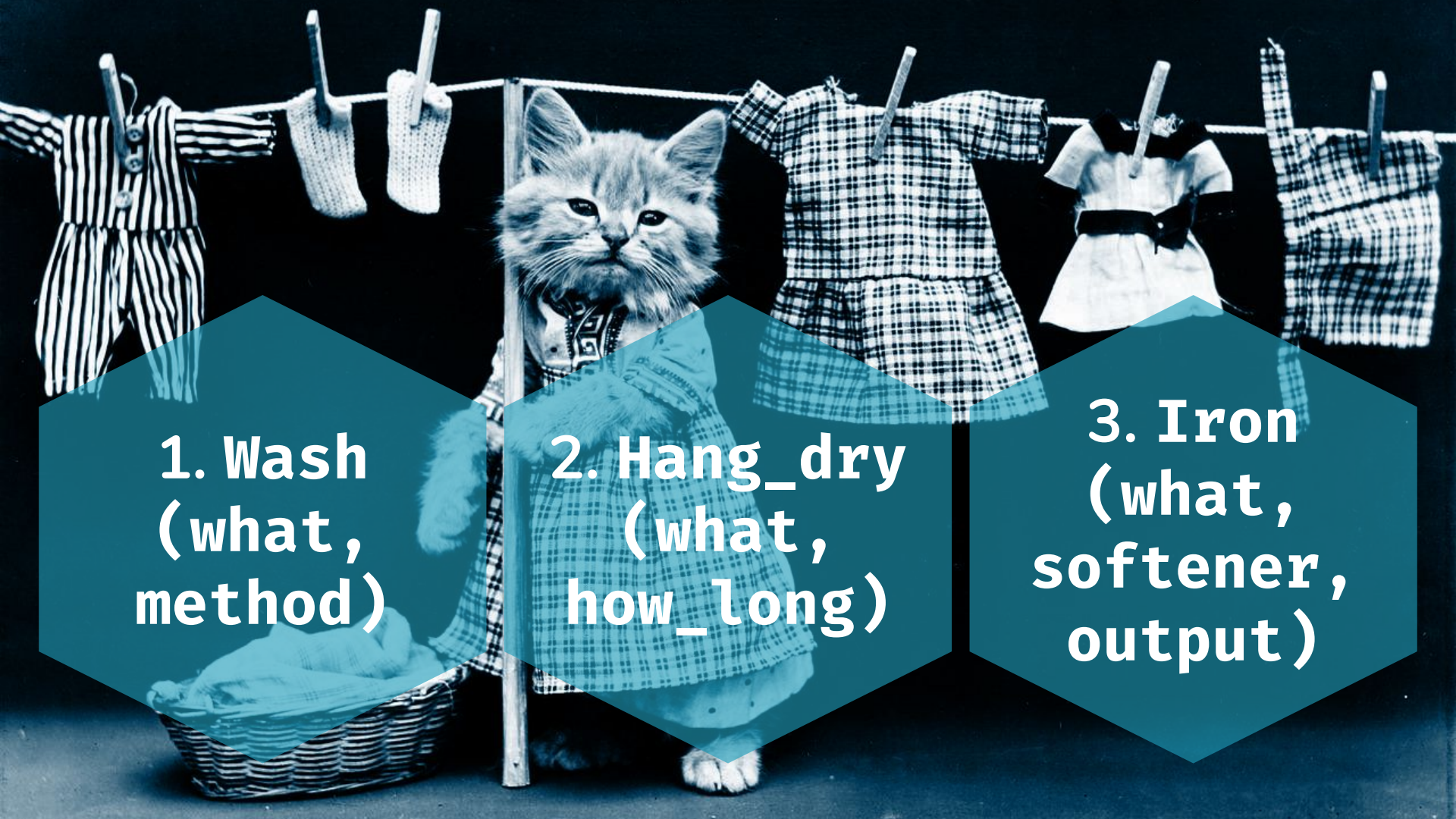
- ! (not)
- | (or)
- & (and)
- ==, !=
- <, <=, >, >=
- %in%
- is.na()

Credits to Michael Toth

How can I

chain?


1. Wash
2. Hang_dry
3. Iron


1. Wash (what, method)
2. Hang_dry (what, how_long)
3. Iron (what, softener, output)


```
> washed_clothes ← wash(what = dirty_clothes,
                        method = “handwash”)
> dry_clothes ← hang_dry(what =
                         washed_clothes,
                         how_long = 8)
> iron(what = dry_clothes, softener = TRUE,
       output = “neat.clothes”)
```


```
> washed_clothes ← wash(what = dirty_clothes,
                        method = “handwash”)
> dry_clothes ← hang_dry(what =
                         washed_clothes,
                         how_long = 8)
> iron(what = dry_clothes, softener = TRUE,
       output = “neat.clothes”)
```


```
> iron(
      hang_dry(
            wash(what = dirty_clothes, method =
                  “handwash”),
            how_long = 8),
      softener = TRUE,
      output = “neat.clothes”)
```

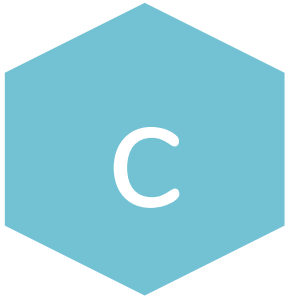
```
function(arg1, arg2, arg3, ...)
```

```
arg1 %>%
  function(arg2, arg3, ...)
```

```
function(arg1, arg2, arg3, ...)
```

```
arg2 %>%
  function(arg1, arg2=., arg3,
           ...)
```

magrittr




```
> wash(what = dirty_clothes,
       method = “handwash”) %>%
     hang_dry(how_long = 8) %>%
          iron(softener = TRUE,
               output = “neat.clothes”)
```

R Syntax Comparison

Dollar sign

goal(data$x, data$y)

- A.k.a base syntax
- Subsetting data by using '[ ]'

Formula

goal(y~x, data=data)

Mostly used in modeling and statistical test

Tidyverse

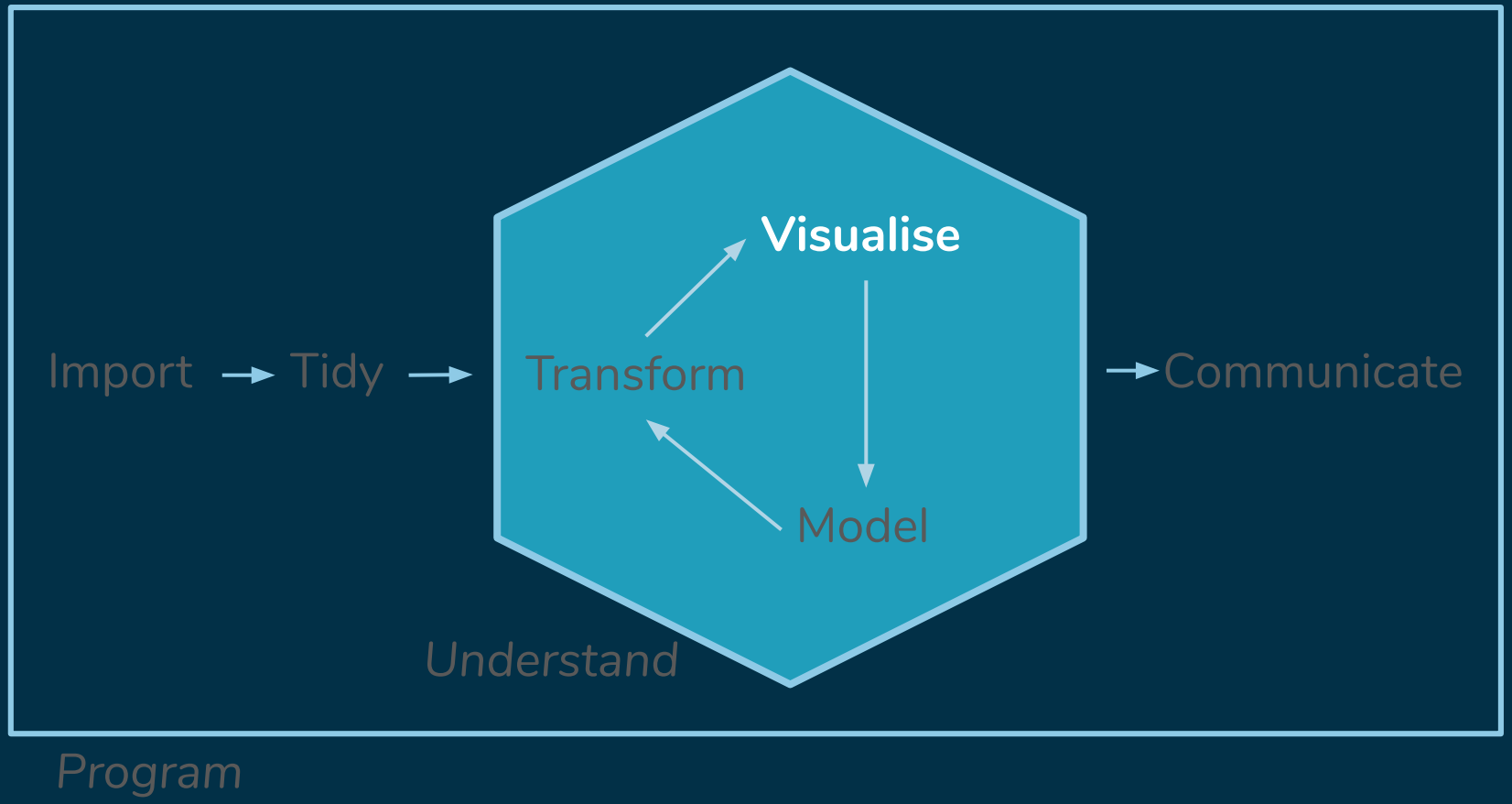
data %>% goal(x, y)

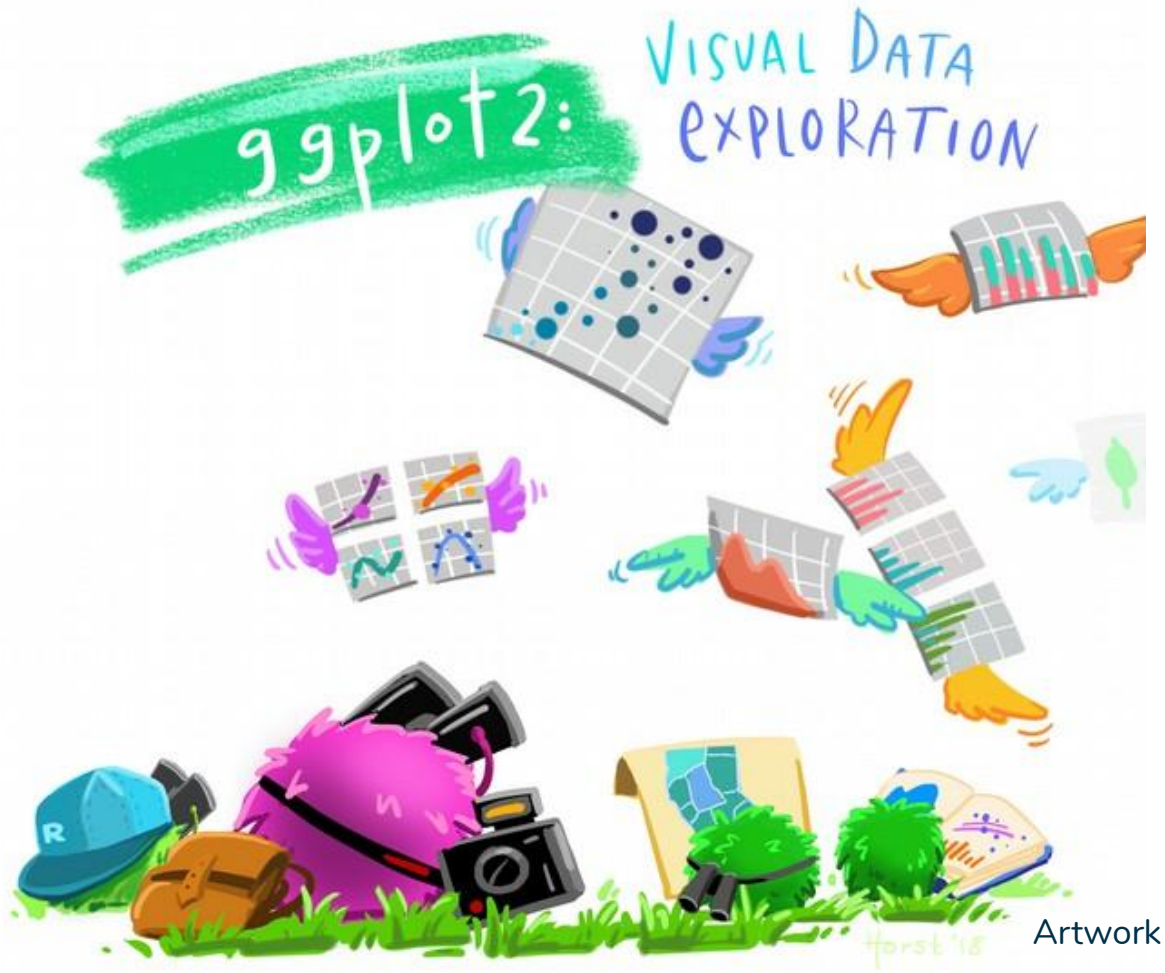
- Expecting data as the first argument
- Plotting using '+' flavour

Cheatsheet: https://github.com/rstudio/cheatsheets/raw/master/syntax.pdf

Let's do **practice**!

- Open 'm3_tidy-data-carpentry.Rmd'
- Do not forget to push your works into GitHub!


Visualise
Import
Tidy
Transform
Communicate
Model
Understand
Program

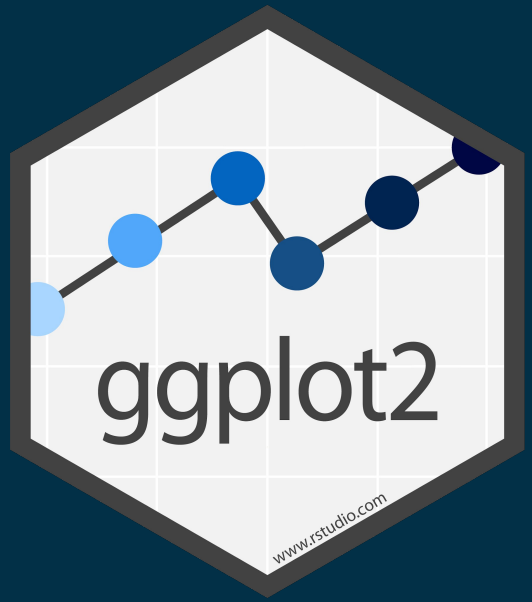


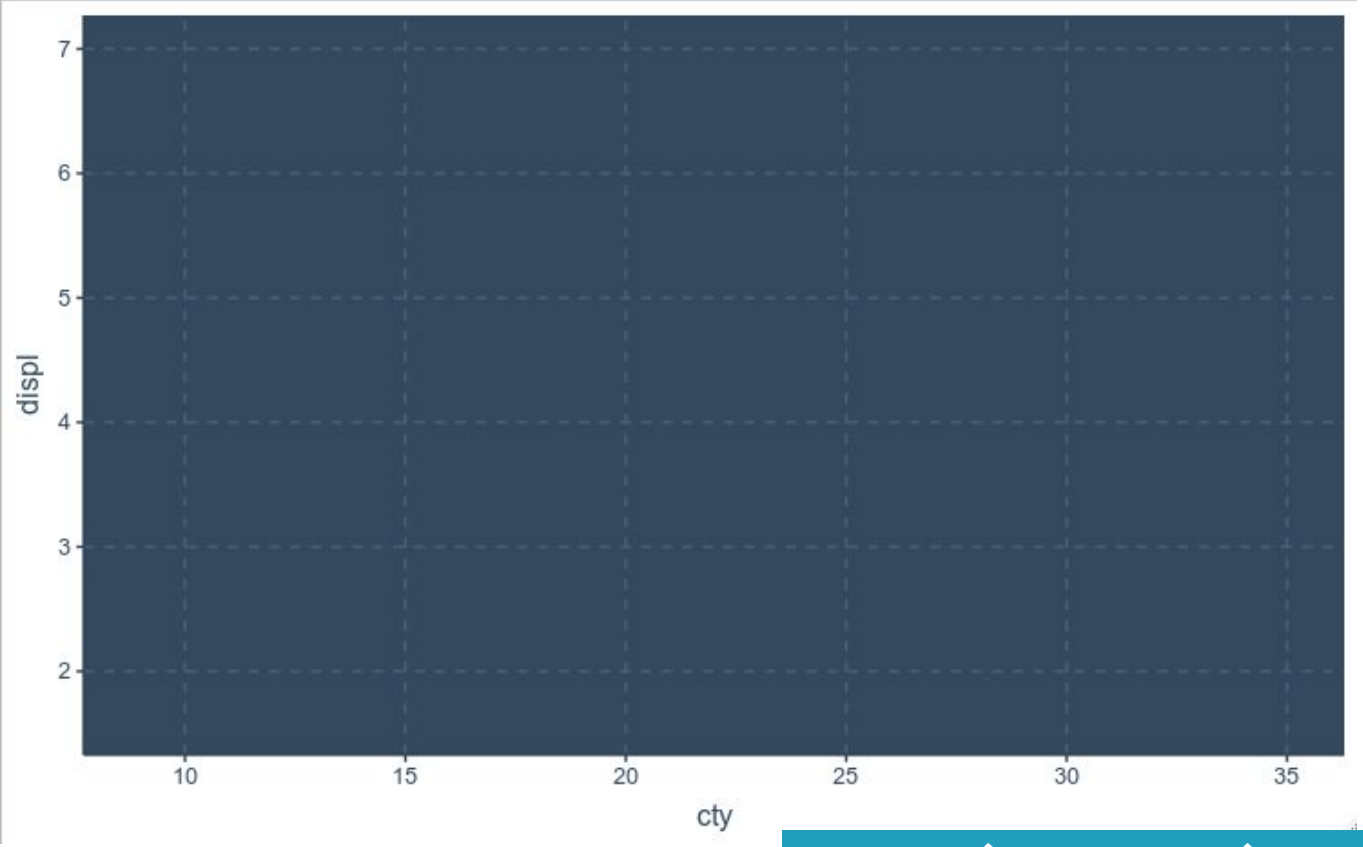
Artwork by @allison_horst

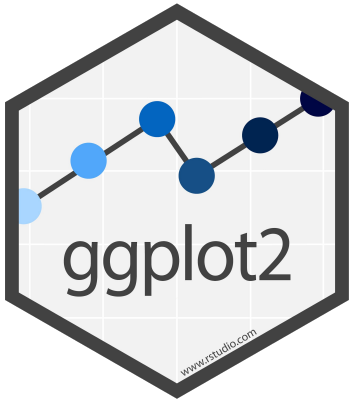
```
ggplot(data) +
    geom_X(mapping=aes(…)) +
    ...

ggplot(data, mapping=aes(...)) +
    geom_X() +
    ...
```

```
data %>%
    ggplot(mapping=aes(...)) +
    geom_X() +
    ...
```

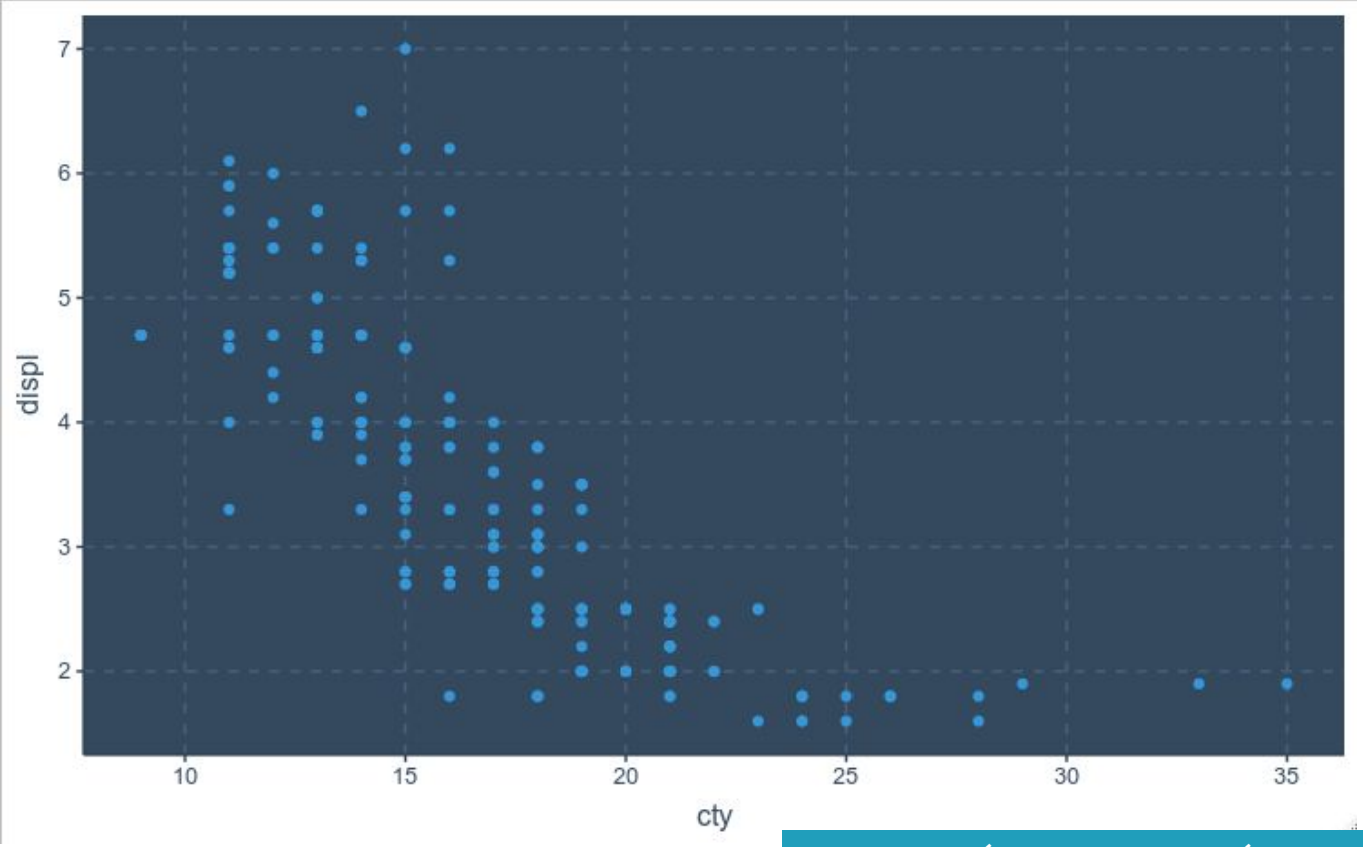
ggplot2

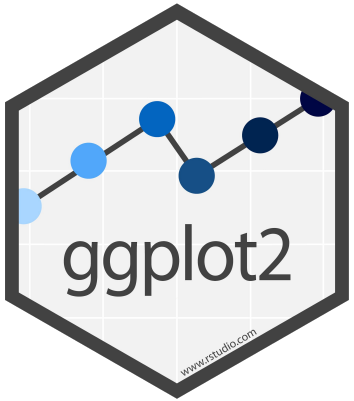




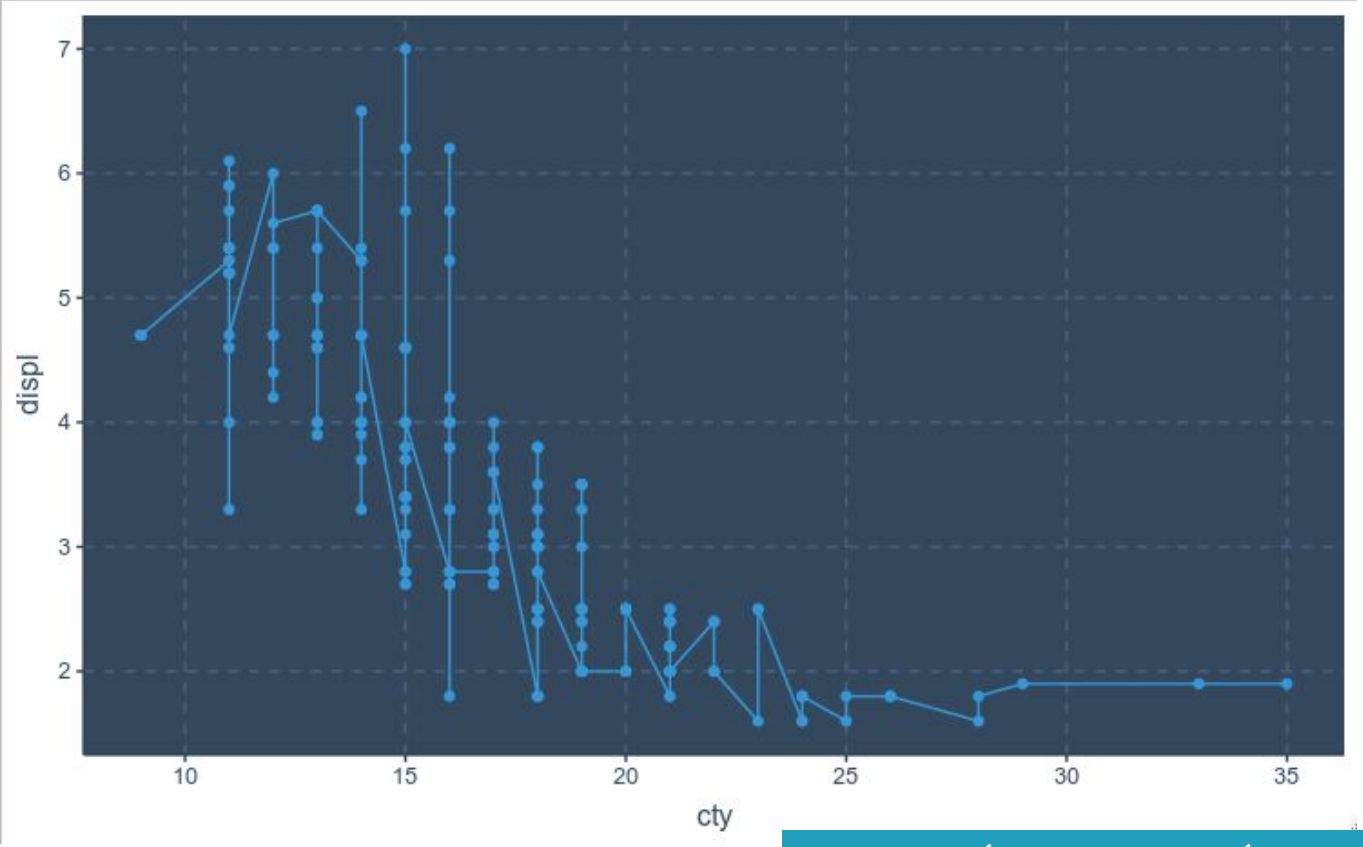


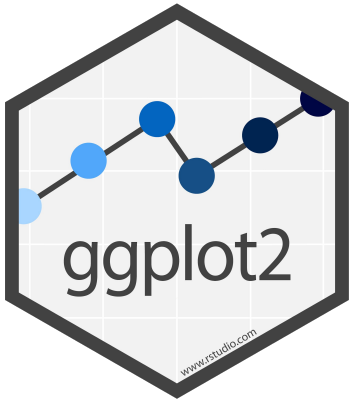

```
ggplot(mpg, aes(x = cty, y = displ))
```



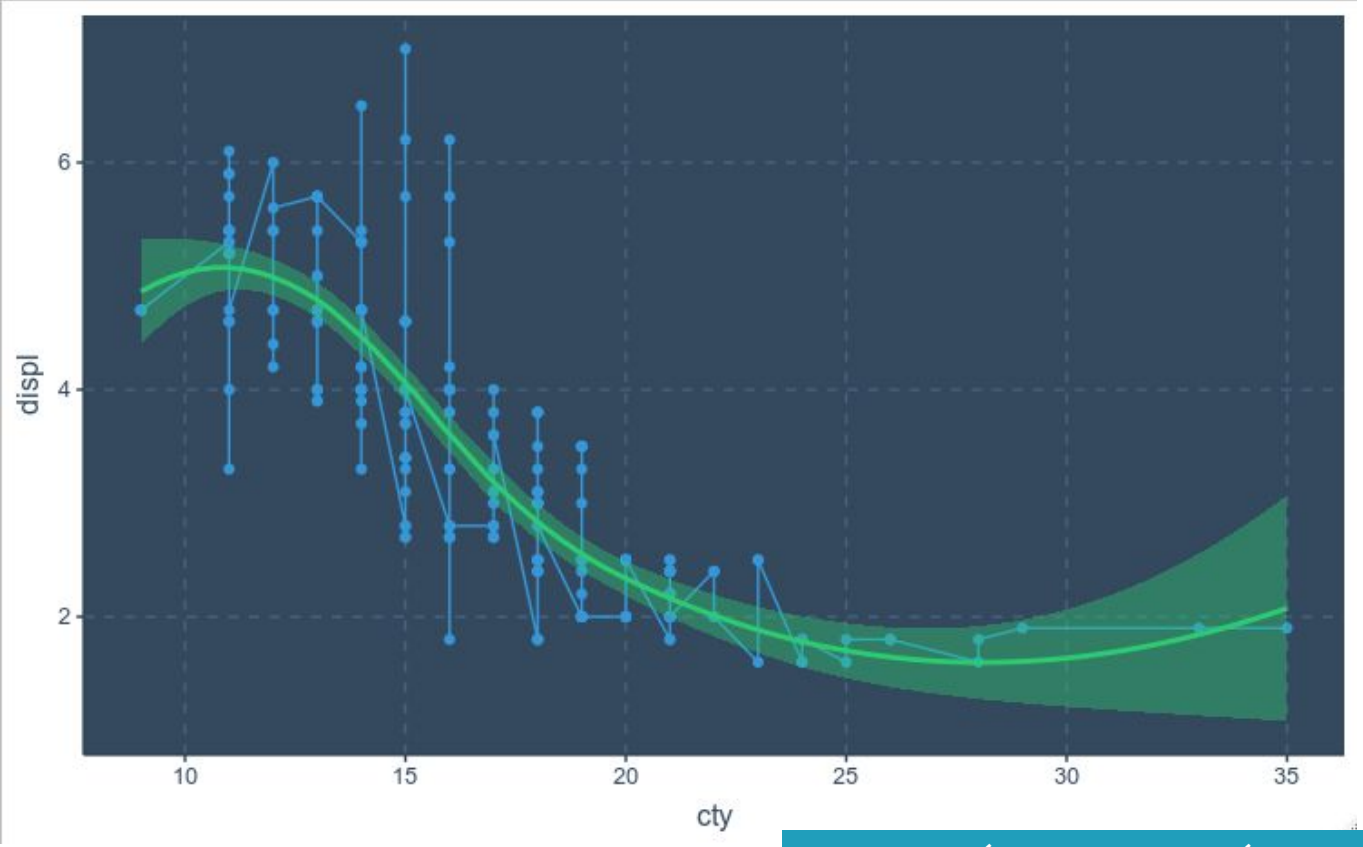


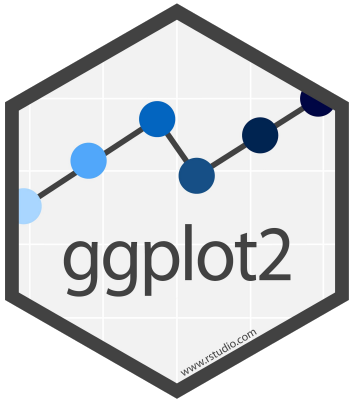

```
ggplot(mpg, aes(x = cty, y = displ)) +
  geom_point()
```






```
ggplot(mpg, aes(x = cty, y = displ)) +
  geom_point() +
  geom_line()
```

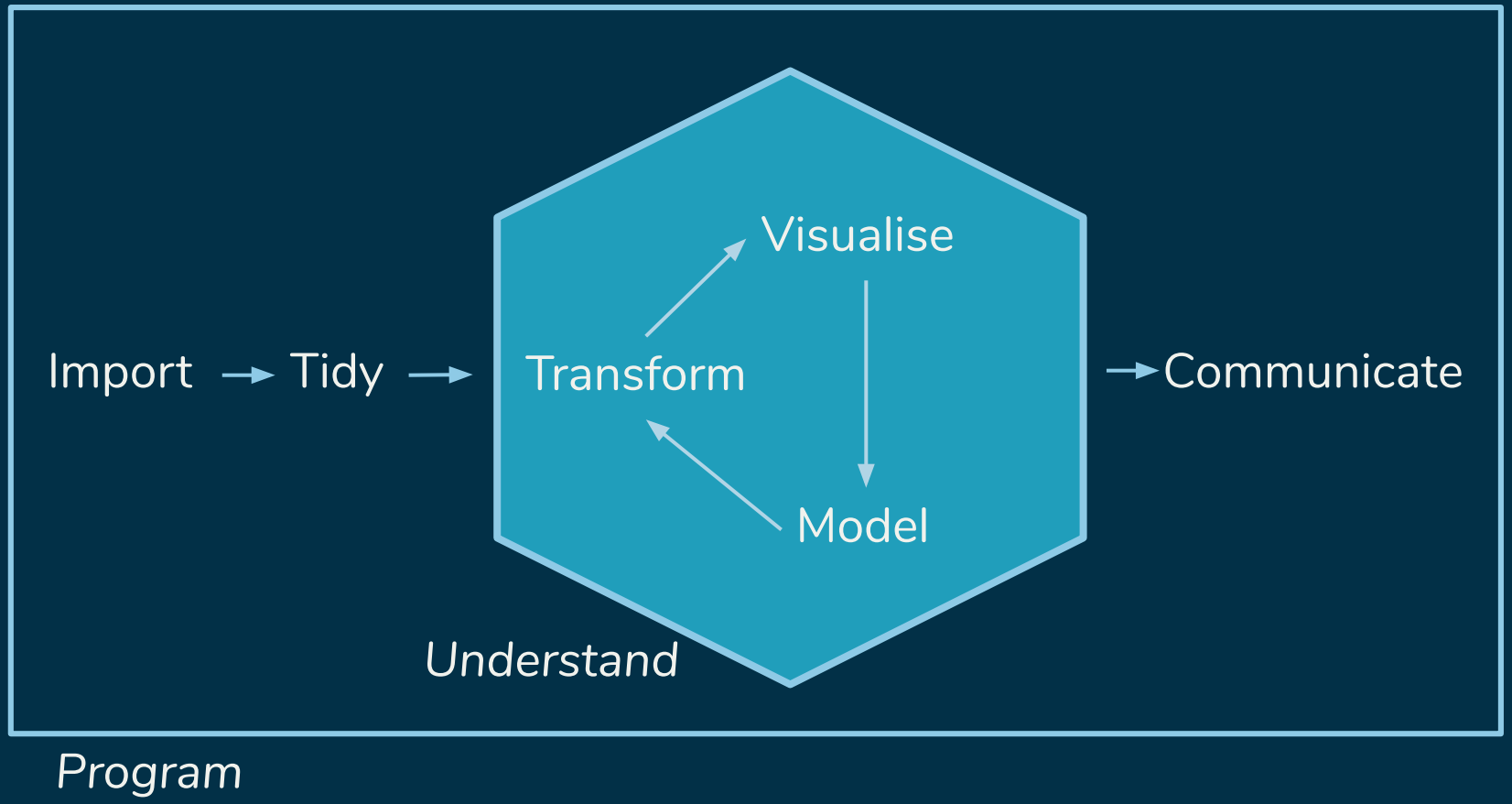


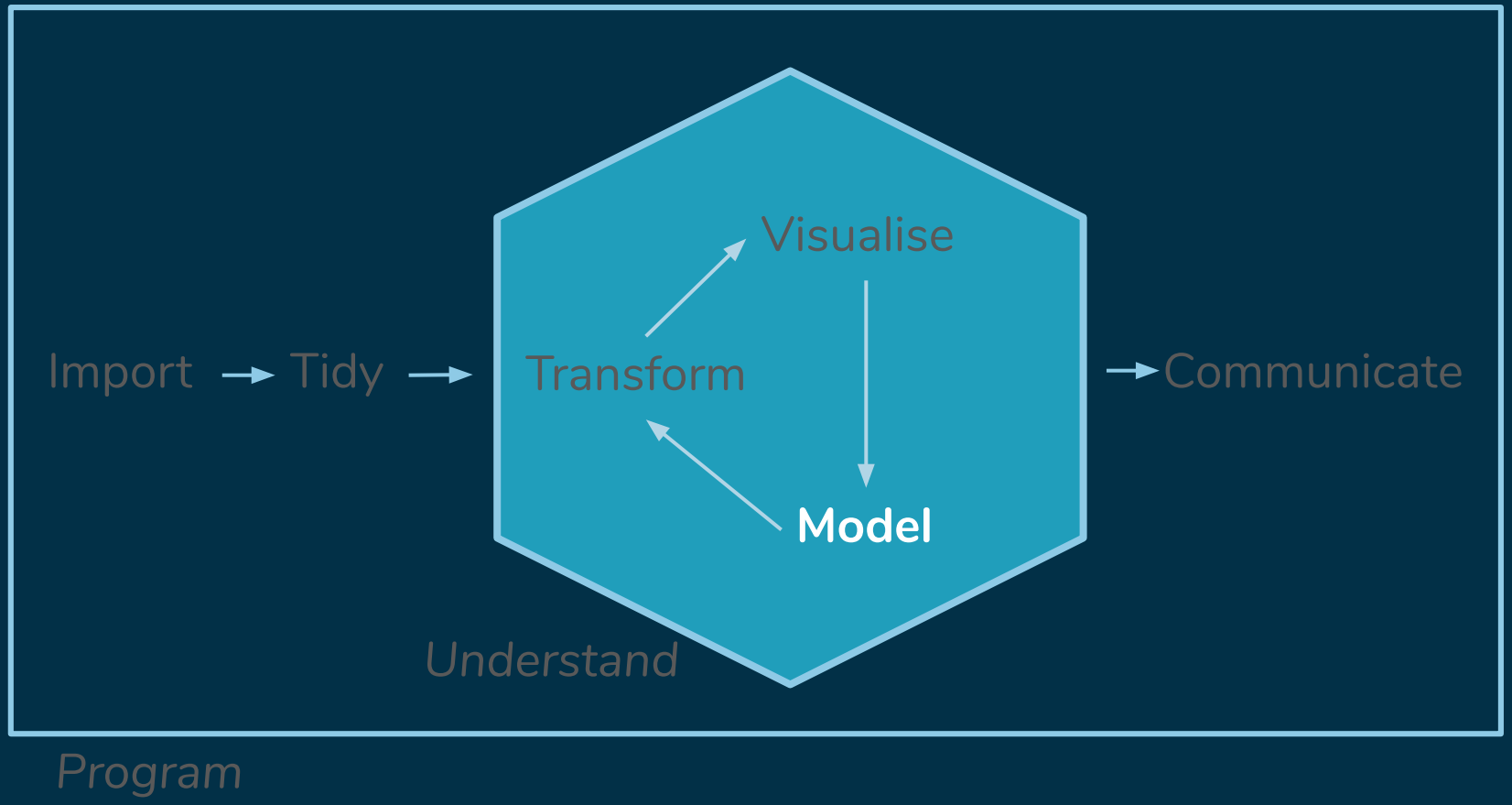



```
ggplot(mpg, aes(x = cty, y = displ)) +
  geom_point() +
  geom_line() +
  geom_smooth()
```

Let's do **practice**!

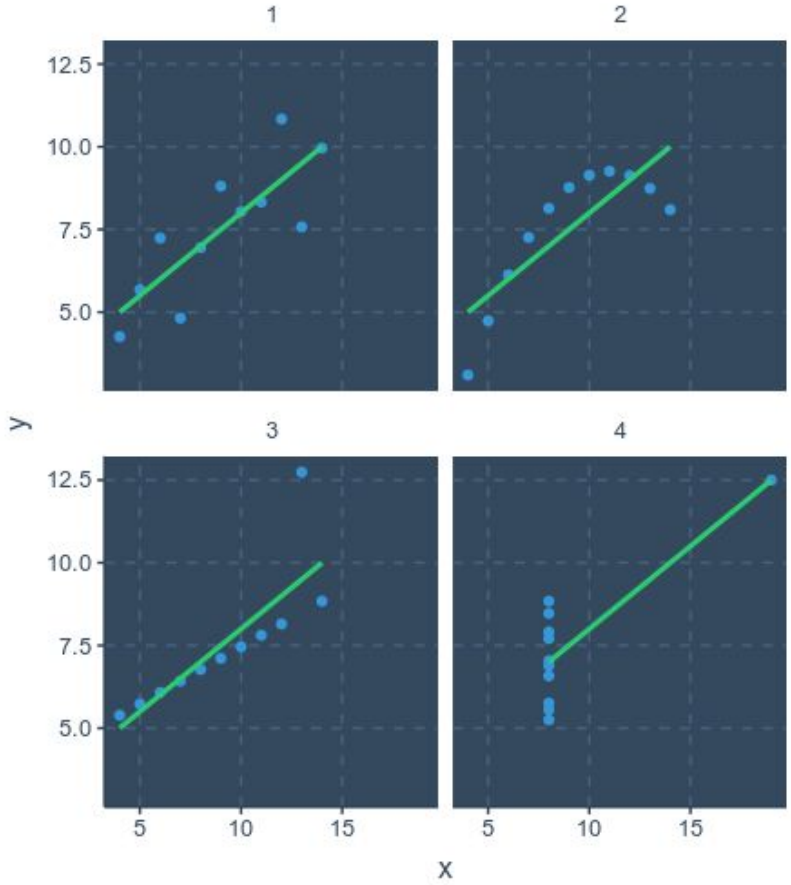
- Open 'm4_data-visualisation.Rmd'
- Do not forget to push your works into GitHub!


Import
Tidy
Transform
Visualise
Model
Communicate
Understand
Program


Import
Tidy
Transform
Visualise
Model
Communicate
Understand
Program

A low dimensional description of a higher dimensional data set

Source: https://github.com/rstudio-education/arm-workshop-rsc2019



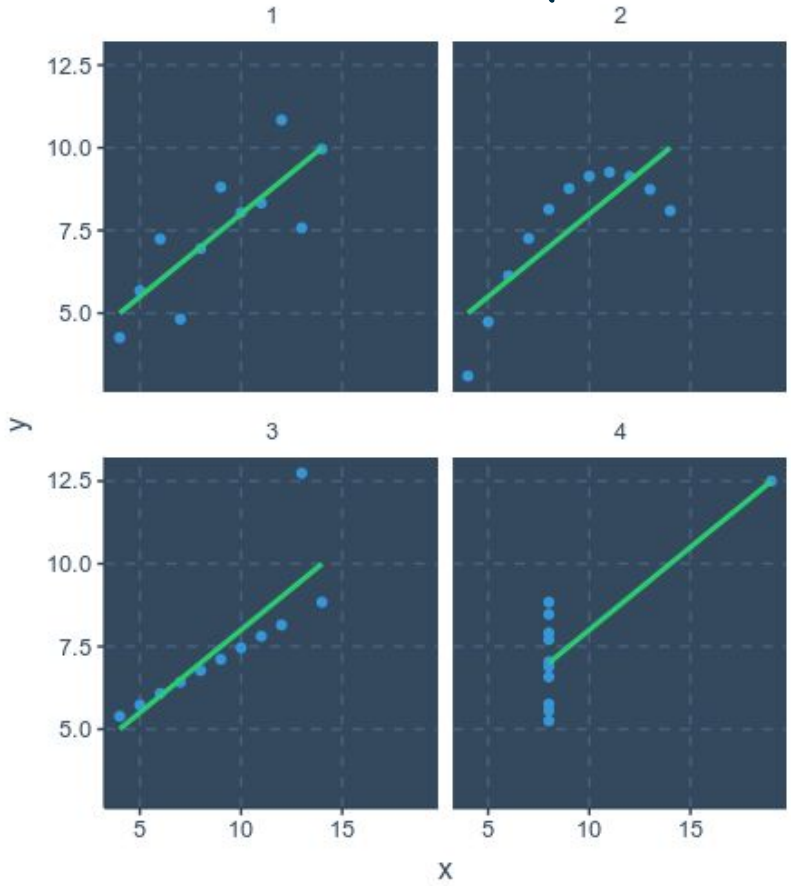
Outcome ~ Predictor/Explanatory

To predict

To explain

- All models are wrong, but some are useful – George Box

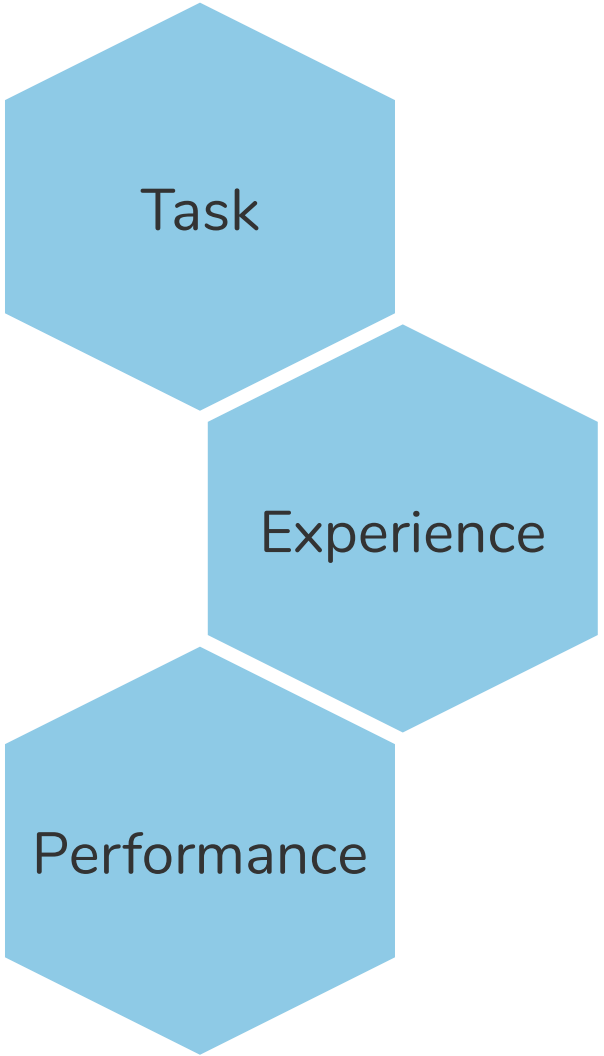
Anscombe's Quartet



Mean of x	9	exact
Sample variance of x	11	exact
Mean of y	7.50	to 2 decimal places
Sample variance of y	4.125	±0.003
Correlation between x and y	0.816	to 3 decimal places
Linear regression line	y = 3.00 + 0.500x	to 2 and 3 decimal places, respectively
R^2	0.67	to 3 decimal places

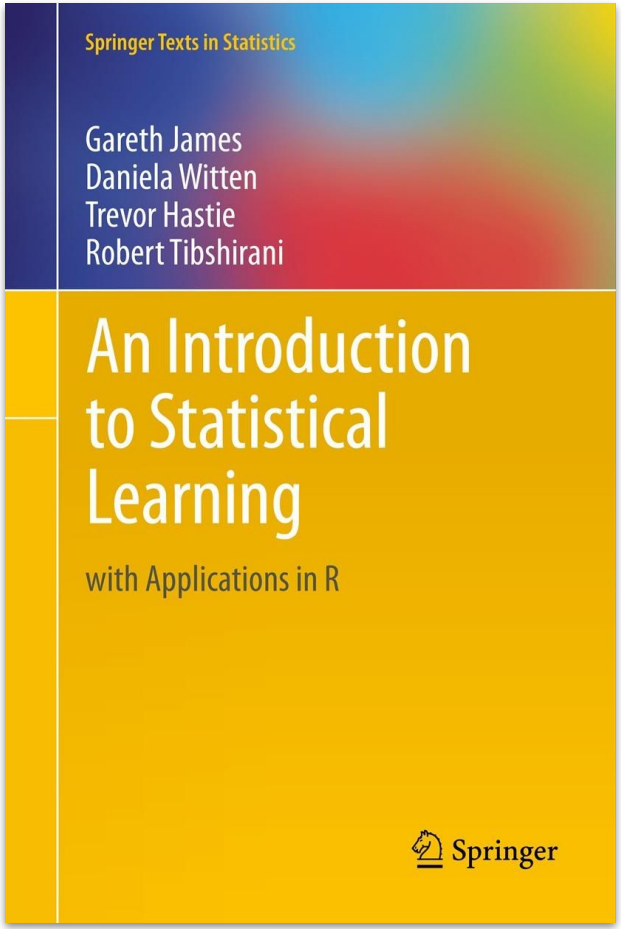
Machine Learning

$$y = f(x) + e$$



Supervised learning

Unsupervised learning



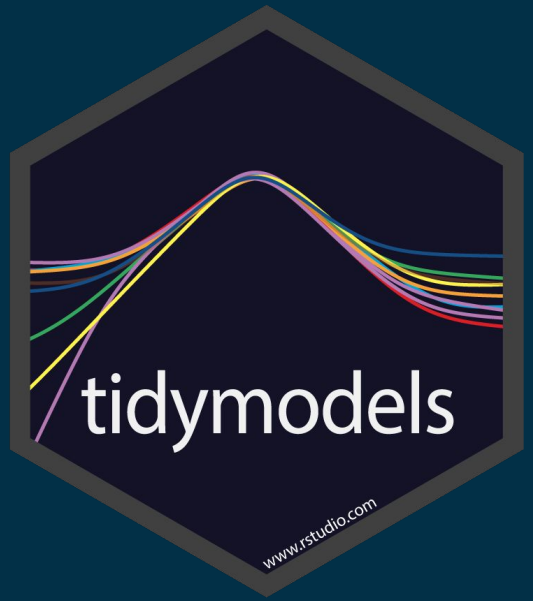
http://faculty.marshall.usc.edu/gareth-james/ISL/

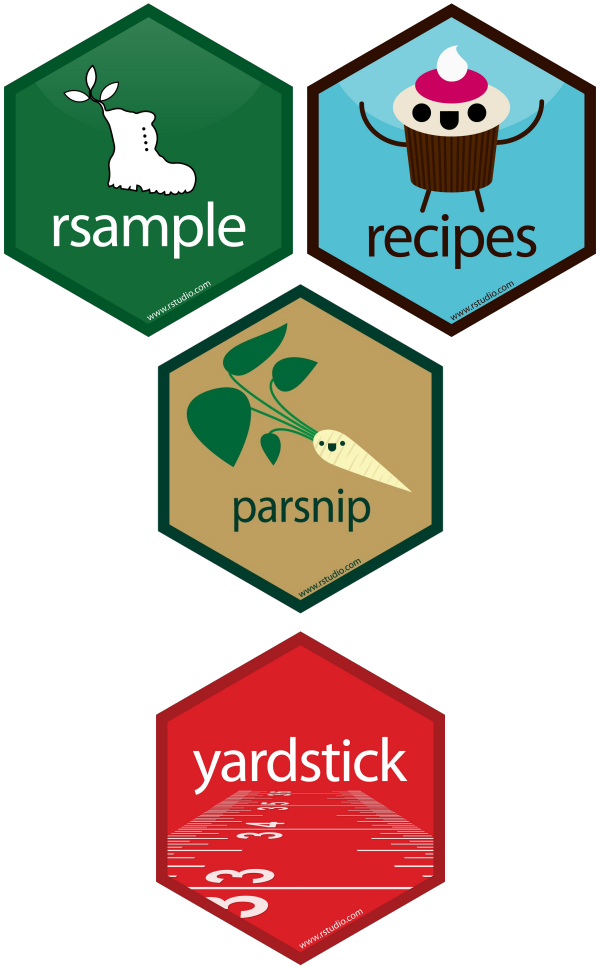
Pre-process

Train

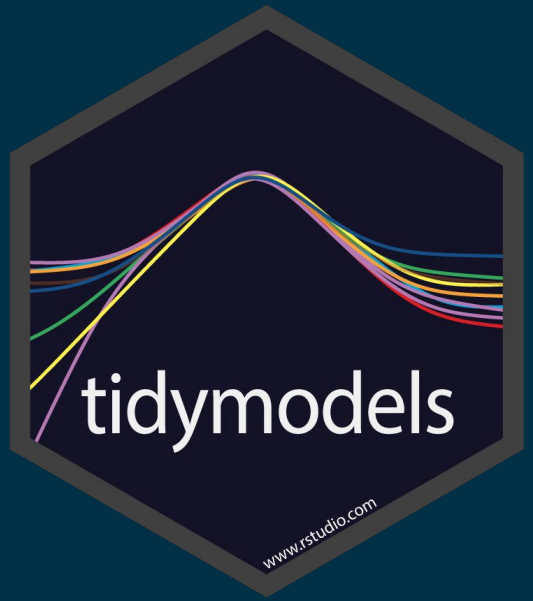
Validate

tidymodels




rsample
www.rstudio.com
recipes
www.rstudio.com
parsnip
www.rstudio.com
yardstick
www.rstudio.com

tidymodels

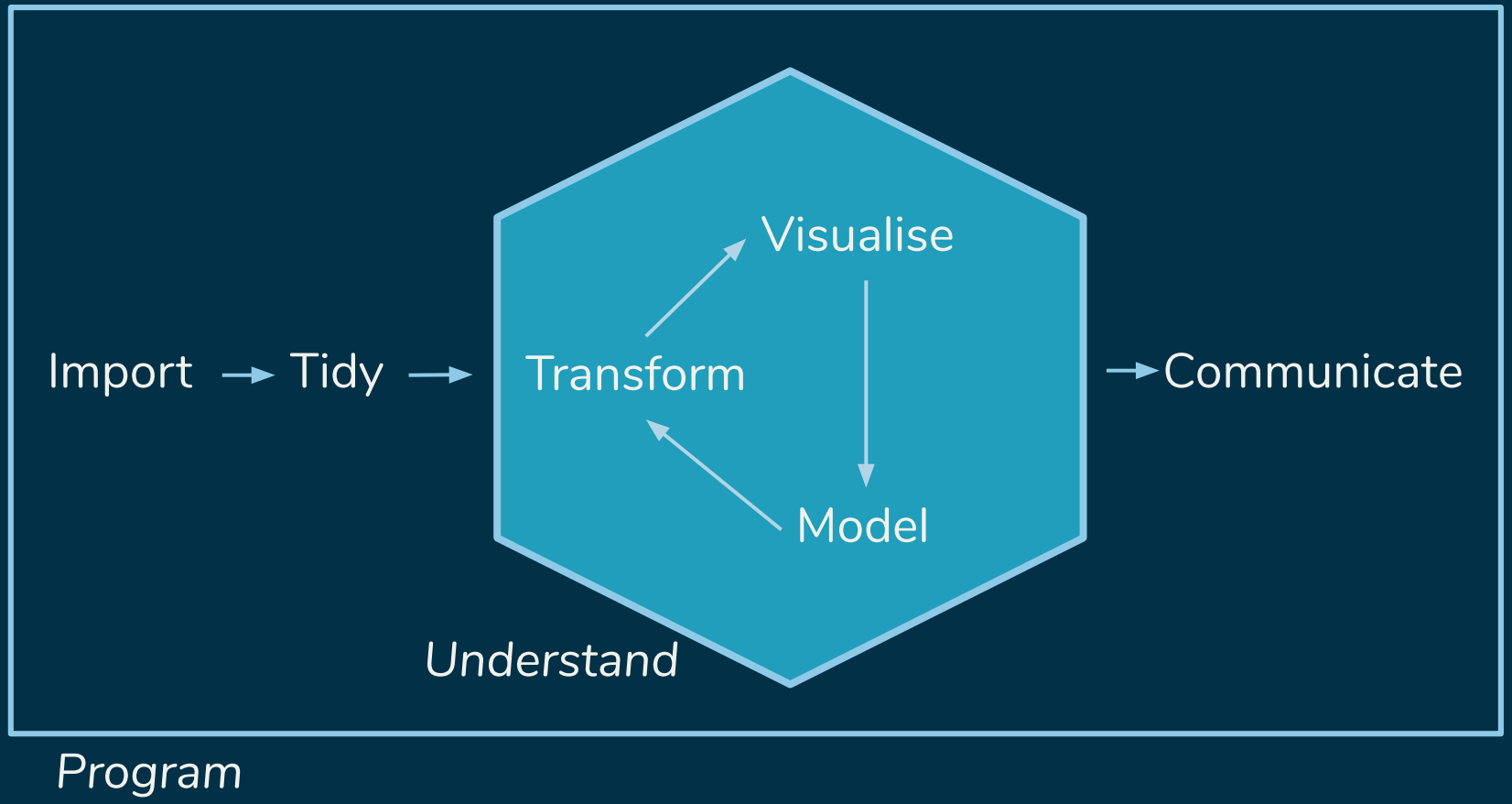

tidymodels
www.rstudio.com

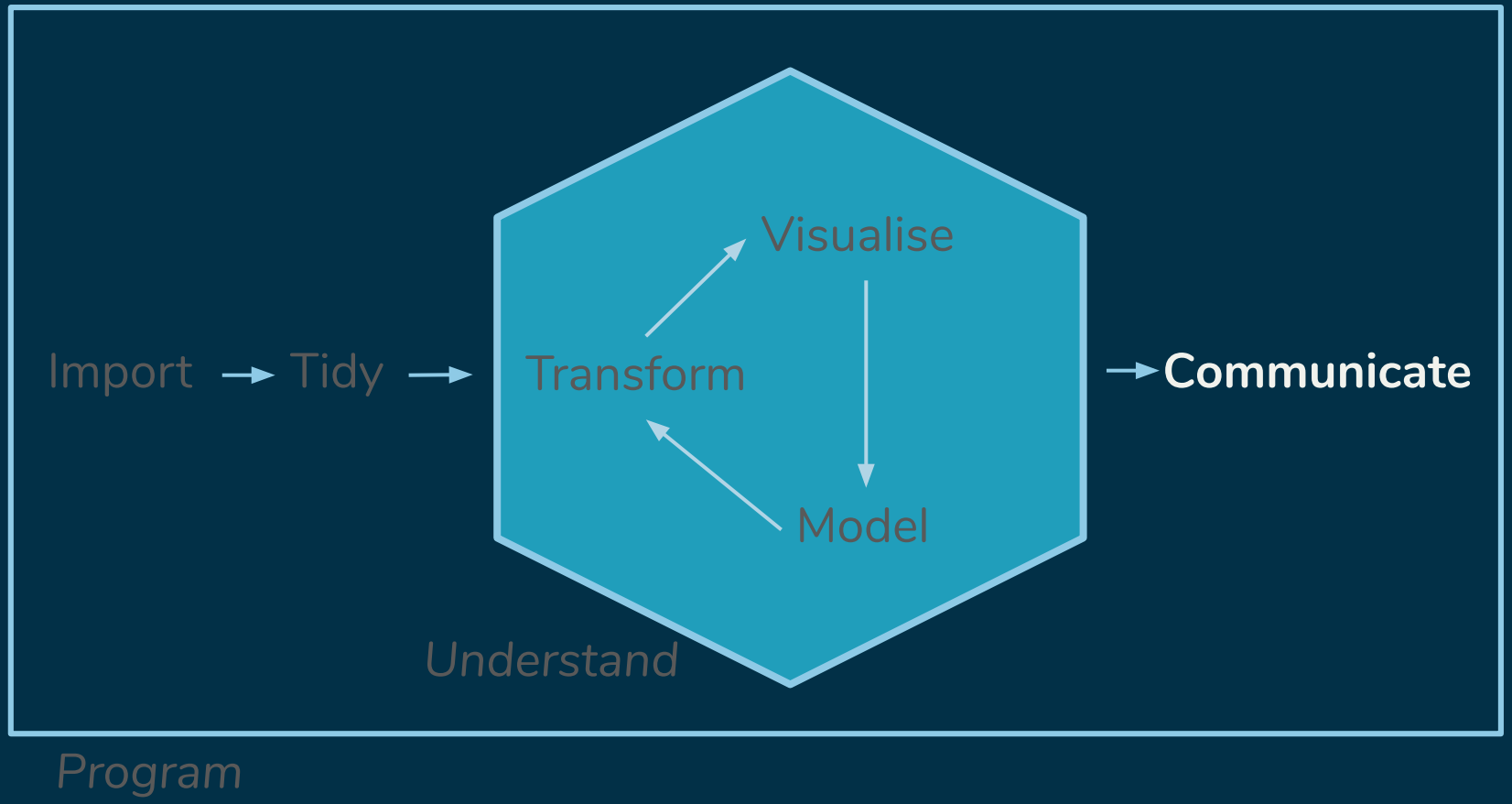
Let's do **practice**!

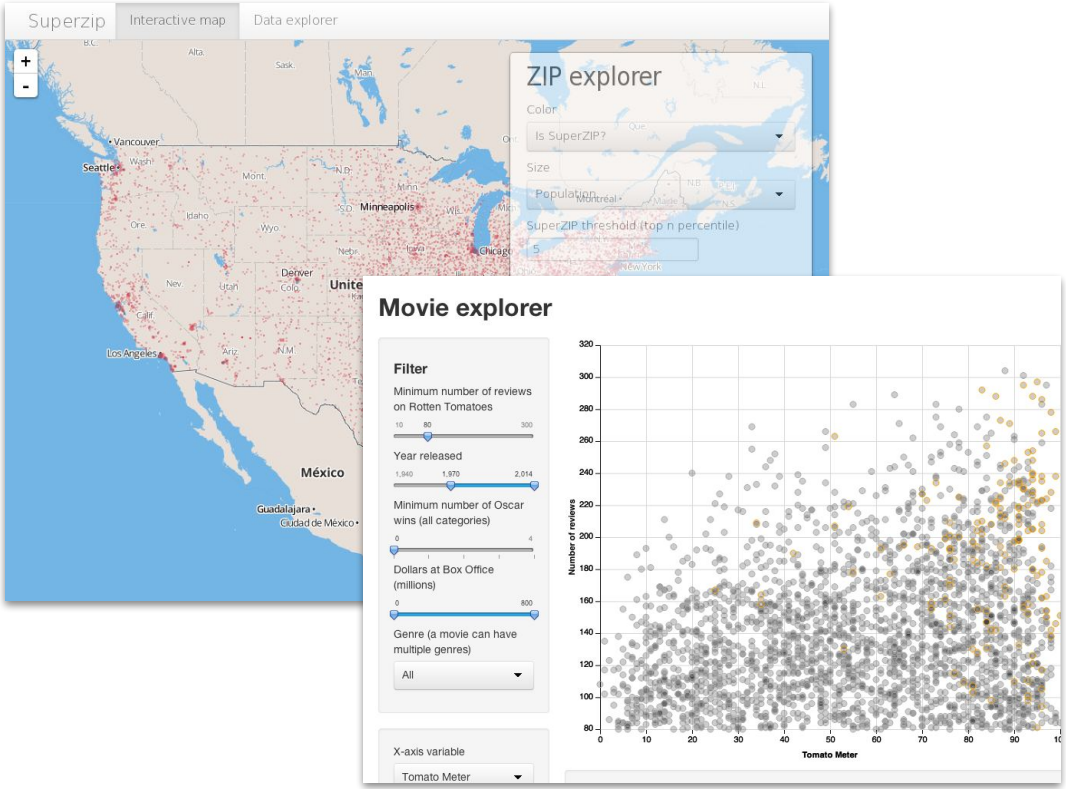
- Open 'm5_model-1.Rmd'
- Do not forget to push your works into GitHub!

Let's do **practice**!

- Open 'm6_model-2.Rmd'
- Do not forget to push your works into GitHub!


Import
Tidy
Transform
Visualise
Model
Communicate
Understand
Program


Import
Tidy
Transform
Visualise
Model
Communicate
Understand
Program



Images from shiny.rstudio.com

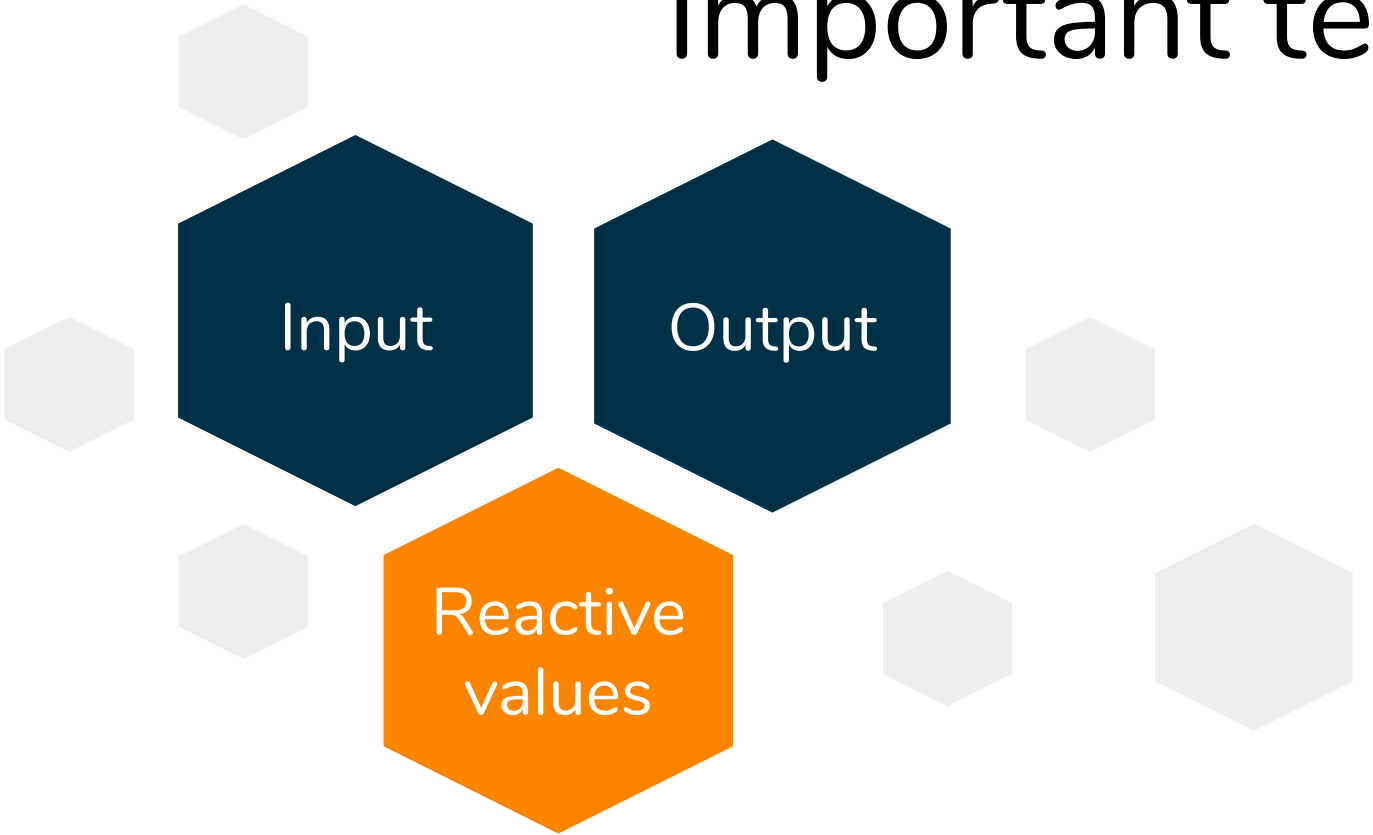
shiny

Shiny

www.rstudio.com

```
library(shiny)
mydata <- load(“mydata.rda”)

ui <- fluidPage(
# define user interface here
)

server <- function(input, output) {
# put logic and computation here
}

shinyApp(ui = ui, server = server)
```

Important terms!



```
library(shiny)
mydata <- load(“mydata.rda”)

ui <- fluidPage(
# define user interface here
)

server <- function(input, output) {
# put logic and computation here
}

shinyApp(ui = ui, server = server)
```

- Inputs are defined and laid out
- Outputs are laid out

- Inputs are taken
- Outputs are calculated/made

```
library(shiny)
mydata <- load("mydata.rda")

ui <- fluidPage(
# define user interface here
)

server <- function(input, output) {
# put logic and computation here
}

shinyApp(ui = ui, server = server)
```

1. Take input value from **input$xx**
2. Perform processing/calculation in reactive context (expression/observer)
3. Build objects to display using **render*()**
4. Save objects to display to **output$xx**

Let's do **practice**!

- Open 'm7_shiny-1.Rmd'
- Do not forget to push your works into GitHub!

Let's do **practice**!

- Open 'm8_shiny-2.Rmd'
- Do not forget to push your works into GitHub!

Congrats!

Artwork by @allison_horst

Contact me

Name: Muhammad Aswan Syahputra
Email: aswansyahputra@sensolution.id
Phone: +62 822 3465 3816
Twitter: @aswansyahputra_